



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

NÁSTROJ PRO PODPORU MANAGEMENTU RIZIK
RISK MANAGEMENT SUPPORT TOOL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Jaroslav Vystavěl

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. Jitka Kreslíková, CSc.

BRNO 2019

Zadání diplomové práce



21645

Student: **Vystavěl Jaroslav, Bc.**
Program: Informační technologie Obor: Management a informační technologie
Název: **Nástroj pro podporu managementu rizik**
Risk Management Support Tool
Kategorie: Softwarové inženýrství
Zadání:

1. Seznamte se se znalostními oblastmi managementu projektů dle aktuálního standardu PMI. Zaměřte se na procesy znalostní oblasti managementu rizik, zejména na plánování a identifikaci rizik.
2. Specifikujte požadavky na systém pro podporu plánování a identifikace rizik v projektech vývoje softwarových produktů. Systém navrhnete.
3. Zvolte vhodné vývojové prostředí a implementujte prototyp funkcí navrženého systému, vybraných po dohodě s vedoucí.
4. Na vzorku dat, vybraném po dohodě s vedoucí, ověřte funkčnost vytvořeného prototypu nástroje.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti jeho dalšího rozšíření.

Literatura:

- *A Guide To The Project Management Body Of Knowledge*: Sixth Edition, Project Management Institute, 2017. ISBN 978-1-62825-184-5.
- SCHWALBE Kathy. *Řízení projektů v IT*: Computer Press, a.s., 2007, 720 s., ISBN 978-80-251-1526-8.
- PANDIAN C. Ravindranath.: *Applied Software Risk Management - A Guide for Software Project Managers*: Auerbach Publications, 2007. ISBN 0-8493-0524-1.
- VOSE David. *Risk Analysis A Quantitative Guide*: John Wiley & Sons, Inc., 2008. ISBN 978-0-470-51284-5.
- SMEJKAL Vladimír, RAIS Karel. *Řízení rizik ve firmách a jiných organizacích*: Grada Publishing, a.s., 2013. ISBN 978-80-247-4644-9.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Kreslíková Jitka, doc. RNDr., CSc.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 16. října 2018

Nástroj pro podporu managementu rizik

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. RNDr. Jitky Kreslíkové, CSc.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jaroslav Vystavěl

19. 5. 2019

Poděkování

Rád bych poděkoval doc. RNDr. Jitce Kreslíkové, CSc. za motivaci a odborné rady, které mi pomohly při tvorbě této práce. Dále bych rád poděkoval doc. Ing. Jaroslavu Zendulkovi, CSc. za cenné rady při tvorbě konceptuálního návrhu systému.

Abstrakt

Tato diplomová práce se zabývá řízením rizik. Celá práce je koncipována dle současného standardu PMI. Jedná se o rozbor deseti znalostních oblastí a procesů v nich obsažených. Po vysvětlení základních pojmů z oblasti projektového řízení jsou všechny procesy stručně popsány včetně jednotlivých vstupů a výstupů každého procesu, které na sebe často navazují. Největší pozornost je soustředěna na oblast řízení rizik, která je předmětem této práce. Z této znalostní oblasti jsou dle zadání podobněji rozebrány procesy plánování a identifikace rizik. Na základě obsahu těchto dvou procesů byl navržen program aplikující zmíněné procesy. Je zde popsán jeho návrh a implementace ve aplikačním rámci PHP Laravel. Následuje testování a závěr, ve kterém jsou diskutovány možná rozšíření.

Abstract

This master thesis deals with risk management. The whole work is conceived according to the current PMI standard. This is an analysis of the ten knowledge areas and the processes involved. After explaining the basic concepts of project management, all processes are briefly described, including the individual inputs and outputs of each process. The most attention is focused on the knowledge area of risk management, which is the subject of this work. From this knowledge area, the risk planning and identification processes are analyzed in more detail according to the assignment. On the basis of these two processes was created an architecture design of application. It describes application design and implementation in PHP framework Laravel. Final part of this thesis consists of testing chapter and final conclusion where the results and possible extensions are discussed.

Klíčová slova

projektové řízení, rizika, PMI, řízení rizik, webová aplikace, Laravel, PHP

Keywords

project management, risk, PMI, risk management, web application, Laravel, PHP

Citace

VYSTAVĚL Jaroslav: Nástroj pro podporu managementu rizik, diplomová práce, Brno, FIT VUT v Brně, 2019. Vedoucí práce doc. RNDr. Jitka Kreslíková, CSc.

Obsah

Obsah	1
1 Úvod.....	2
2 Představení problematiky projektového řízení	3
2.1 Projektové řízení dle PMI.....	3
2.2 Znalostní oblasti a procesy	5
2.3 Řízení integrace projektu.....	5
2.4 Řízení rozsahu projektu.....	8
2.5 Řízení plánování projektu.....	10
2.6 Řízení ceny projektu.....	13
2.7 Řízení kvality projektu	14
2.8 Řízení zdrojů projektu.....	16
2.9 Řízení komunikace v rámci projektu.....	19
2.10 Řízení obstarávání v rámci projektu.....	21
2.11 Řízení zúčastněných stran.....	22
2.12 Řízení rizik projektu.....	24
3 Specifikace požadavků a návrh	33
3.1 Analýza požadavků pro identifikaci rizik.....	34
3.2 Návrh schématu databáze	36
3.3 Diagram případů užití a identifikace aktérů	38
4 Implementace.....	42
4.1 Výběr aplikačního rámce.....	42
4.2 Popis implementace grafického rozhraní	45
4.3 Popis implementace případů užití	47
5 Testování	57
5.1 Verifikační testování	57
5.2 Uživatelské testování.....	57
5.3 Závěry testování.....	60
6 Závěr	61
7 Seznam příloh na CD	65

1 Úvod

Řízení rizik je jednou z nejkritičtějších oblastí vedení projektů. Riziko je vyjádřením neurčitosti a nejistoty. V podstatě každá lidská činnost je riziková a lidé si často ani neuvědomují, že řízení rizik je na jejich denním pořádku. Každý člověk si podvědomě vytváří protiopatření před riziky, případně rizikové a havarijní plány, aby snížil dopad případné nepříznivé situace. Analogicky by se toto mělo dít v projektovém řízení. Otázka správně sestavené a dodržované rizikové politiky často souvisí s úspěchem projektu, potažmo firmy jako takové. K řízení rizik je tedy nutné přistupovat zodpovědně, protože v mnoha případech může rozdíl mezi úspěšně dokončeným projektem a projektem neúspěšným tkvět v zanedbaném risk managementu.

Tato práce začíná vysvětlením základních pojmů projektového řízení v kapitole Představení problematiky projektového řízení. V této poměrně rozsáhlé kapitole je dále popsáno deset znalostních oblastí dle současného standardu PMI z roku 2017 včetně procesů, které obsahují. Detailněji jsou popsány procesy z oblasti rizik, zejména první dva, které se týkají identifikace a plánování rizik.

Kapitola Specifikace požadavků a návrh obsahuje návrh aplikace, který sestává z modelu případů užití, jenž byl navržen na základě předchozí specifikace požadavků na aplikaci plynoucí ze získaných teoretických znalostí. Je zde popsán příslušný diagram případů užití včetně detailního strukturovaného popisu vybraného případu užití. Dále je zde popsáno schéma databáze, která je použita.

V kapitole Implementace je proveden rozbor aplikačních rámců PHP následovaný detailnějším popisem některých charakteristických rysů vybraného aplikačního rámce Laravel. Poté je popsán způsob jakým bylo implementováno uživatelské rozhraní – zde je nastíněno jakým způsobem jsem realizoval jednotlivé prvky grafického uživatelského rozhraní. Dále následuje detailnější popis implementace případů užití, tedy jak k ní bylo přistupováno v rámci použití architektury MVC (model-view-controller). Z této podkapitoly by mělo být patrné jaký vztah mezi sebou mají jednotlivé moduly projektu.

Testování aplikace je popsáno v samostatné kapitole, která řeší způsoby a pojednává o výsledcích testování. Mimo verifikačního testování uživatelského rozhraní a aplikační logiky je proveden popis jednotlivých přístupů k testování, představen profil testerů a podrobně popsán způsob provádění uživatelského testování. V závěru testování je popsán dopad, který byl způsoben touto etapou na podobu výsledného produktu. Poslední kapitola Závěr je věnovaná zhodnocení práce a možnostem rozšíření stávajícího řešení.

Práce navazuje na stejnojmenný semestrální projekt.

2 Představení problematiky projektového řízení

Ještě před samotným vysvětlením obsahu projektového řízení je vhodné si ujasnit jak je v kontextu této práce chápán pojem *projekt*. Projektem se rozumí *časově ohraničené úsilí vynaložené k vytvoření jedinečného produktu nebo služby*. Již samotná definice obsahuje několik klíčových aspektů, které by měl každý projekt splňovat. Jedním z nich je časová ohraničenost – jde o dočasné snažení, které vrcholí tím, že je např. dosažen cíl projektu, cíle nemůže být dosaženo nebo cíl již není potřebný. Jedinečnost projektu spočívá v tom, že se skládá z kombinace procesů, která je svým způsobem unikátní a pravděpodobně se nebude v této podobě již nikdy opakovat.

Projektové řízení je *aplikací znalostí, dovedností, nástrojů a technik při realizaci projektových aktivit za účelem dosažení požadavků projektu*. Účelem existence této oblasti managementu je zlepšení vedení projektů, jejich organizace, alokace zdrojů, komunikační stránka, dodržení stanoveného rozpočtu, analýza a opatření rizik, které vedou k úspěšnému dosažení cíle, nebo alespoň je tímto postupem snížena pravděpodobnost neúspěchu projektu.

2.1 Projektové řízení dle PMI

PMI (Project Management Institute) je nezisková organizace sídlící v USA, která vydala publikaci s názvem *A Guide To The Project Management Body Of Knowledge*. Jedná se o metodologickou příručku a souhrn nejlepších postupů (angl. best-practices) primárně určenou pro projektové manažery. Veškeré informace v této práci se zakládají na skutečnostech popsanych v šestém vydání této knihy vydané v září roku 2017. Procesy, postupy a metody uvedené v tomto svazku vycházejí z praxe odborníků z celého světa [6]. Tímto je tedy zaručeno, že nejde o pouhé teoretické pokusy popsat praxi, nýbrž o reálné zkušenosti a praktiky používané v organizacích po celém světě.

PMI definuje deset znalostních oblastí:

- řízení integrace projektu,
- řízení rozsahu projektu,
- řízení plánování projektu,
- řízení ceny projektu,
- řízení kvality projektu,
- řízení zdrojů projektu,
- řízení komunikace v rámci projektu,
- řízení rizik projektu,
- řízení obstarávání v rámci projektu a
- řízení zainteresovaných stran.

Do každé z deseti oblastí může náležet proces z tzv. skupin procesů. Je definováno pět skupin:

- iniciační,
- plánovací,
- realizační,
- monitorovací a ovládací a
- ukončovací procesy.

Skupiny procesů jsou logickou kategorizací procesů. Všechny procesy z vybrané skupiny se snaží dosáhnout určitého společného cíle. PMI popisuje v rámci své publikace 47 procesů rozdělených do výše zmíněných skupin. Je vhodné říci, že více než polovina všech

procesů – 24 – náleží do plánovací skupiny procesů. Množství procesů, které de facto patří do plánovací etapy projektu, jen podtrhuje důležitost této fáze projektu.

Vztah mezi těmito dvěma stěžejními prvky (skupiny procesů a znalostní oblasti) je zachycen na obrázku Obr. 1. Z tabulky na tomto obrázku je dobře rozpoznatelné, který proces náleží které oblasti a které skupině procesů. Pojem *proces* je dle PMI definován jako „*systematická posloupnost aktivit směřujících k výsledku takovému, který je přetransformováním vstupů na výstupy*“. V kontextu projektového řízení je proces chápán jako komponenta projektu. Každý proces přeměňuje pomocí stanovených řídicích nástrojů vstupy na patřičné výstupy, které mohou sloužit dalším procesům jako vstup, případně se může jednat o výsledek projektu jako celku.

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring and Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Work 4.4 Manage Project Knowledge	4.5 Monitor and Control Project Work 4.6 Perform Integrated Change Control	4.7 Close Project or Phase
5. Project Scope Management		5.1 Plan Scope Management 5.2 Collect Requirements 5.3 Define Scope 5.4 Create WBS		5.5 Validate Scope 5.6 Control Scope	
6. Project Schedule Management		6.1 Plan Schedule Management 6.2 Define Activities 6.3 Sequence Activities 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Plan Cost Management 7.2 Estimate Costs 7.3 Determine Budget		7.4 Control Costs	
8. Project Quality Management		8.1 Plan Quality Management	8.2 Manage Quality	8.3 Control Quality	
9. Project Resource Management		9.1 Plan Resource Management 9.2 Estimate Activity Resources	9.3 Acquire Resources 9.4 Develop Team 9.5 Manage Team	9.6 Control Resources	
10. Project Communications Management		10.1 Plan Communications Management	10.2 Manage Communications	10.3 Monitor Communications	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses	11.6 Implement Risk Responses	11.7 Monitor Risks	
12. Project Procurement Management		12.1 Plan Procurement Management	12.2 Conduct Procurements	12.3 Control Procurements	
13. Project Stakeholder Management	13.1 Identify Stakeholders	13.2 Plan Stakeholder Engagement	13.3 Manage Stakeholder Engagement	13.4 Monitor Stakeholder Engagement	

Obr.1: Kategorizace procesů do znalostních oblastí dle PMI (převzato z [1], s. 25)

2.2 Znalostní oblasti a procesy

Jsou svým způsobem dalším rozdělením procesů do kategorií. Znalostní oblast je oblast projektového řízení, která je definována znalostními požadavky a popsána procesy, postupy, vstupy, výstupy, nástroji a technikami, které do ní patří. V tomto textu jsou u jednotlivých procesů v rámci každé z oblastí zmíněny některé z možných vstupů a výstupů, je nutné zdůraznit, že tyto dvě množiny se u konkrétních projektů mohou zásadně lišit. Vstupy a výstupy jsou popsány pouze pro uvedení čtenáře do obrazu, nejedná se v žádném případě o přesné požadavky. Tato variantnost je způsobena různým charakterem řízených projektů (velikost firmy/projektového týmu, odvětví, množství zúčastněných stran, typ produktu/služby/cíle, atd.). Mezi typické představitele vstupních a výstupních elementů patří následující:

- *projektové dokumenty* – může se jednat např. o odhad nákladů, registr rizik, seznam projektových činností, atd. Prakticky jde o souhrn dokumentů, které jsou výstupem z jednotlivých procesů.
- *plán řízení projektu* – tento termín je poněkud zavádějící, protože se nejedná o jeden konkrétní plán, ale je to prakticky soubor plánů vytvořených v různých znalostních oblastech (např. plán řízení kvality, plán řízení rizik, plán řízení kvality, cenový odhad, atd.).
- *faktory podnikového prostředí* - jsou podmínky a omezení, která působí na tvůrčí tým, potažmo na průběh projektu, zevnitř společnost a jsou důsledkem jejího vnitřního klimatu.
- *OPA* – je zkratkou pro *organizational process assets* a jedná se o plány, postupy, procesy a znalostní báze, které si uchovává společnost ve svém interním vlastnictví a typicky je nezveřejňuje.

2.3 Řízení integrace projektu

Řízení integrace projektu zahrnuje procesy a aktivity, které umožňují řídit a koordinovat různé komponenty, ze kterých projekt sestává. Jako jediná ze všech znalostních oblastí zasahuje polem působnosti do všech pěti skupin procesů. Do této oblasti náleží 7 procesů.

2.3.1 Vytvoření základací listiny projektu

Zakládací listina je klíčovým dokumentem při vzniku projektu, ve kterém je stanoven projektový manažer. Osoba uvedená v této listině na této pozici má kompetenci zacházet se zdroji společnosti a alokovat je pro potřeby projektu.

Vstupy:

- obchodní dokumenty,
- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- základací listina projektu.

2.3.2 Řízení prací na projektu

Proces, který obsahuje samotné provádění aktivit, které byly v předchozí fázi definované v plánu řízení projektu, který je také jedním ze vstupů tohoto procesu. Primárním účelem

tohoto procesu je zvýšení šancí na zdárné dokončení projektu, kterého může být dosaženo pouze zodpovědným řízením a kontrolou.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- odsouhlasené požadavky na změnu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- cíle projektu,
- data o výkonu práce,
- seznam požadavků na změnu,
- upravený plán řízení projektu,
- upravené další projektové dokumenty,
- OPA.

2.3.3 Řízení projektových znalostí

Smyslem tohoto procesu je využití existujících znalostí společnosti/týmu ke snadnějšímu dosažení vytyčeného cíle projektu. Dále pak využití znalostí a zkušeností nabytých při řešení současného projektu v projektech budoucích. Současně také může být provedena změna v plánu řízení, která může být způsobena vznikem nových poznatků.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- cíle projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- rejstřík získaných poznatků,
- upravený plán řízení,
- upravené OPA.

2.3.4 Monitorování a kontrola projektových prací

Jde o proces sledování, posuzování a hlášení celkového postupu prací na projektu, který by měl vést k dosažení předem určených cílů v plánu řízení projektu. Tímto způsobem mohou mít zúčastněné strany přehled o tom, co už bylo provedeno a v jaké fázi se projekt nachází. Pokud se v průběhu projektu vyskytne nějaká překážka, je zachycena právě tímto procesem. V neposlední řadě tento proces umožňuje zúčastněným stranám pozorovat, zda se proces nevychyluje z pohledu finančního odhadu a časového plánu.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- informace o pracovním výkonu,

- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- zpráva o pracovním výkonu,
- seznam požadavků změn,
- upravený plán řízení,
- upravené další projektové dokumenty.

2.3.5 Provádění kontroly změn

Tímto procesem se rozumí navrhování změn, jejich schvalování, řízení změn ve vztahu k cíli projektu. Ve své podstatě se tedy jedná o proces, který zajistí, že případné změny nebudou mít negativní dopad na průběh projektu. Toho je dosaženo pečlivým zvážením rizik souvisejících se změnou a jejich případným dopadem.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- zpráva o výkonu práce,
- seznam požadavků na změnu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- odsouhlasené požadavky na změnu,
- upravený plán řízení projekt,
- upravené další související projektové dokumenty.

2.3.6 Uzavření projektu nebo jeho dílčí fáze

Jde o proces ukončení všech aktivit a činností projektu nebo fáze. Naplánovaná práce je dokončena a týmy, které se podílely jsou uvolněny a mohou být zařazeny do jiného projektu. Výstupem je mimo jiné finální zpráva o projektu a další dokumenty související s jeho průběhem, které mohou být společností využity při realizaci dalších projektů.

Vstupy:

- zakládací listina projektu,
- plán řízení projektu,
- další projektové dokumenty,
- přijaté cíle,
- obchodní dokumenty,
- smlouvy,
- dokumentace obstarávání projektu,
- OPA.

Výstupy:

- upravené další související projektové dokumenty,

- upravený rejstřík získaných poznatků,
- finální zpráva,
- upravené OPA,
- předání produktu/služby/výsledku projektu.

2.4 Řízení rozsahu projektu

Ve znalostní oblasti řízení rozsahu projektu jsou obsaženy procesy, které zajistí, že práce vykonaná na projektu bude dostačující k pokrytí stanovených cílů. Zároveň by mělo být zajištěno i to, že nebude vykonávána zbytečná práce. Oblast obsahuje celkem šest procesů, které náleží do plánovací a monitorovací skupiny, z nichž většina probíhá již ve fázi plánovací. Je to poměrně logické, vzhledem k tomu, že pro úspěch projektu je klíčová právě tato etapa. V rámci kontextu je nutno rozlišovat dva pojmy:

- rozsah projektu – jedná se o vynaložené úsilí k dosažení cíle projektu a
- rozsah produktu – souhrn funkcí a vlastností produktu.

2.4.1 Plánování řízení rozsahu

Jedná se o proces vytváření plánu rozsahu, který zachycuje jakým způsobem bude rozsah projektu a produktu definován, validován a kontrolován. Smyslem tohoto procesu je mít přehled nad rozsahem v průběhu provádění projektových prací.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení rozsahu,
- plán řízení požadavků.

2.4.2 Sběr požadavků

Je proces určování, dokumentace a řízení zájmů a požadavků zúčastněných stran. Tento postup zajistí, že je možné správně specifikovat rozsah produktu, resp. projektu. Finální podoba značně závisí na odvětví, ve kterém se projekt pohybuje. Požadavky na produkt v odvětví těžkého strojírenství budou zcela jistě odlišné od těch v informačních technologiích. Typickým nástrojem pro sběr požadavků jsou diskuze, rozhovory a dotazníky.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- projektové dokumenty,
- obchodní dokumenty,
- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- dokumentace požadavků,
- matice stopovatelnosti požadavků.

2.4.3 Definování rozsahu

Definování rozsahu je procesem vytváření podrobného popisu produktu či projektu. Postup je takový, že je převzata dokumentace požadavků z procesu Sběr požadavků a na základě této je vytvořen detailní popis produktu (služby).

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- projektové dokumenty (dokumentace požadavků),
- faktory podnikového prostředí,
- OPA.

Výstupy:

- stanovení rozsahu projektu,
- úpravy příslušných dokumentů.

2.4.4 Vytváření WBS

WBS je zkratkou pro termín *work breakdown structure*, jde o hierarchickou dekompozici celkového rozsahu práce nutné pro dosažení cíle. Jedná se tedy o rozdělení celkového objemu pracovních úkonů do menších, dílčích, celků, které jsou lépe zvladatelné z pohledu řízení. K vytvoření WBS je nutné perfektně znát požadavky na produkt a mít k dispozici dokument o rozsahu plánované práce.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- shrnutí rozsahu projektu (komplexní informace ohledně WBS a struktury prací),
- úpravy příslušných dokumentů.

2.4.5 Ověřování rozsahu

Je proces formálního schválení rozsahu projektu. Přínosem tohoto postupu je, že bude dodán takový produkt, jaký zákazník požaduje. Aby toho bylo dosaženo, je nutné tento proces provádět opakovaně periodicky v průběhu projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- ověřené cíle,
- data o výkonosti práce.

Výstupy:

- přijaté cíle,
- informace o výkonu práce,
- seznam požadavků na změnu,
- upravené příslušné související dokumenty.

2.4.6 Kontrola rozsahu

Je proces monitorování rozsahu produktu a projektu včetně kontroly, zda není překračována výchozí hodnota rozsahu. Stejně jako v předchozím případě se jedná o proces, který je nutné opakovat periodicky, pokud se má dostavit jeho pozitivní účinek. V průběhu tohoto procesu musí být zajištěno, že všechny změny, které by se měly odehrát, jsou součástí procesu Provádění kontroly změn, který již ze své podstaty, dohlíží na to, aby nedošlo k narušení integrity projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- informace o výkonu práce,
- seznam požadavků na změnu,
- upravený plán řízení projektu,
- upravené příslušné související dokumenty.

2.5 Řízení plánování projektu

Třetí znalostní oblastí nazvaná řízení plánování projektu řeší časovou stránku projektu, resp. jeho řízení z hlediska času (tj. rozvrh). Smyslem této oblasti je mj. vytvoření plánu projektu, který pomáhá zúčastněným získat přehled o tom kdy a jak bude dodán výsledný produkt/služba. Oblast obsahuje šest procesů, přičemž pět z nich patří do plánovací skupiny. V této sekci je také vhodné zmínit agilní přístup, který je zde často aplikován. Agilní přístup využívá plánování na kratší časové úseky, kdy je vykonána naplánovaná práce, posléze je zhodnocena a případné nedostatky jsou ihned opraveny.

2.5.1 Řízení časového plánu projektu

Je proces vytváření zásad, postupů a dokumentace pro plánování, vývoj, vedení, provádění a kontrolu projektového rozvrhu. Tento postup vede k ujasnění si přístupů jakými bude vedeno provádění časového rozvrhu.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- časový plán projektu.

2.5.2 Definování činností

Definování činností je procesem identifikace a dokumentace aktivit, které musí být provedeny pro dosažení cíle projektu. V průběhu procesu dochází k dekompozici objemu práce do jednotlivých činností, se kterými se lépe pracuje z hlediska časového odhadu práce nutné pro provedení.

Vstupy:

- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam činností,
- seznam vlastností činností,
- seznam milníků,
- seznam požadavků na změny,
- upravený plán řízení projektu.

2.5.3 Posloupnost činností

V rámci tohoto procesu jsou identifikovány a popsány vztahy mezi činnostmi projektu. Na základě těchto poznatků jsou jednotlivé činnosti seřazeny do posloupnosti, ve které budou vykonány a to tak, aby bylo dosaženo co největší efektivity práce.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- síťový graf návaznosti činností,
- upravené projektové dokumenty.

2.5.4 Odhad časové náročnosti

Činnostem, které byly v předchozích procesech identifikovány, popsány a zdokumentovány, jsou přiřazeny doby trvání. Jedním z přístupů k výpočtu této doby je metoda 3 možných hodnot: pesimistická, optimistická a realistická, ze kterých je vypočten aritmetický průměr, jehož hodnota se uvažuje jako finální pro danou činnost.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,

- OPA.

Výstupy:

- odhady trvání,
- seznam faktů pro odhad dob trvání,
- upravené projektové dokumenty.

2.5.5 Tvorba časového rozvrhu

Je proces analýzy posloupnosti činností, dob jejich trvání, zdrojů potřebných k jejich provedení, časových omezení, které vedou k vytvoření časového plánu. Tento plán, ve kterém jsou uvedena i datová rozmezí jednotlivých aktivit, je poté využit při samotném provádění projektu. Zajímavým prvkem, který se při plánování využívá, je metoda kritické cesty (zkr. CPM), kdy jsou činnosti včetně jejich časových náročností vyobrazeny ve formě síťového grafu, ve kterém je poté nalezena nejdelší (tzv. kritická) cesta z pohledu trvání. Kritická cesta značí, že činnosti, které na ní leží se nesmí zpozdit oproti plánu, pokud se tak stane, nastane i posun termínu ukončení celého projektu. Je to způsobeno tím, že oproti ostatním činnostem, které na ni neležejí, mají tyto nulovou časovou rezervu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- smlouvy
- faktory podnikového prostředí,
- OPA.

Výstupy:

- základní časové schéma,
- časový plán projektu,
- další podružné informace k časovému plánu,
- projektový kalendář,
- seznam požadavků na změny,
- upravený plán řízení,
- upravené související projektové dokumenty.

2.5.6 Kontrola časového rozvrhu

Jde o proces kontroly časového plánu v průběhu projektu. Provádění projektu je sledováno a vyhodnocováno, v případě nesrovnalostí plánu s realitou jsou podniknuty příslušné kroky a je proveden zásah do časového rozvrhu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- data o výkonu práce,
- OPA.

Výstupy:

- informace o výkonu práce,
- předpověď časového rozvrhu,

- seznam požadavků na změny,
- upravený plán řízení projektu,
- úpravy souvisejících projektových dokumentů.

2.6 Řízení ceny projektu

Ve znalostní oblasti řízení ceny projektu jsou zahrnuty veškeré činnosti související s finanční stránkou projektu. Jsou zde tedy obsaženy procesy plánování, odhadu, financování a řízení ceny. Primárním účelem je udržet projekt ve vymezeném rozpočtu a vyhnout se jeho překročení. Podobně jako v případě předchozích znalostních oblastí je většina procesů soustředěna do plánovací etapy (plánovací skupiny procesů). Při plánování ceny je klíčové přihlídnout k rozsahu projektu. Je evidentní, že pokud není plně znám rozsah projektu, bude obtížné vytvořit cenový odhad. Je tedy nutné, aby zúčastněné strany s tímto faktem počítaly. Pro procesy z této skupiny je charakteristické, že neprobíhají sériově za sebou, nýbrž zároveň, a současně vyžadují výstupy z ostatních procesů jiných znalostních oblastí.

2.6.1 Plánování řízení nákladů

Jedná se o analogii procesu řízení časového plánu projektu v rovině nákladů. V rámci tohoto procesu je tedy určeno jakým způsobem budou náklady na projekt odhadnuty, vyčleněny, řízeny a kontrolovány. Tímto způsobem jsou ujasněny principy při rozhodování o ceně při provádění projektu. Do tohoto procesu vstupuje mj. i plán řízení rizik, který bude charakterizován v kapitole Plánování řízení rizik.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení nákladů.

2.6.2 Odhadování nákladů

Jde o proces odhadu ceny výsledného projektu. Smyslem je stanovení pravděpodobných nákladů na zdroje nutných k průběhu činností. Tento odhad vychází z informací, které jsou k dispozici v určitém časovém okamžiku. V rámci procesu by také mělo dojít ke zvážení zda je nutné příslušné zdroje pořizovat, vyrábět či řešit jejich potřebu způsobem pronájmu, leasingu a případně vybrat optimální možnost. Pro každý projekt je typické, že čím blíže se nachází ke zdárnému ukončení, je i odhad celkové ceny přesnější. Ceny mohou být uvedeny v jednotkách peněžních měn, ale je vhodnější používat jednotky vztažené k lidskému výkonu (člověkohodiny, člověkodny, atd.), aby se odhad vyhnul konfrontaci s peněžními jevy, jako je např. inflace.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- odhad cen,
- seznam faktů pro odhad cen,
- úpravy dalších souvisejících projektových dokumentů.

2.6.3 Stanovení rozpočtu

Proces zahrnuje sumarizaci nákladů jednotlivých projektových činností. Tímto je dosaženo stanovení výchozího rozpočtu za projekt. Tato hodnota je v dalších fázích projektu neustále kontrolována a je hlídáno, zda se náklady neodlišují od očekávané částky.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- obchodní dokumenty,
- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- odhad rozpočtu,
- požadavky na financování,
- úpravy dalších souvisejících projektových dokumentů.

2.6.4 Kontrola nákladů

Kontrola nákladů je proces zabývající se dohledem nad stavem projektu z pohledu nákladů a z pohledu finančních změn vzhledem k předem stanoveným nákladům. Pokud je nutná změna rozpočtu je ji nutno řešit pomocí příslušného procesu - Provádění kontroly změn.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- požadavky na financování,
- data o výkonu práce,
- OPA.

Výstupy:

- informace o výkonu práce,
- předpověď nákladů,
- seznam požadavků na změny,
- úpravy plánu řízení projektu,
- úpravy dalších souvisejících projektových dokumentů.

2.7 Řízení kvality projektu

Pátá znalostní oblast se týká řízení kvality. Obsahuje procesy plánování, řízení a kontroly kvality projektu a produktů dle požadavků zúčastněných stran. Obsahuje tři procesy, které náleží do tří skupin procesů (plánovací, realizační a monitorovací). Řízení kvality je důležité

pro všechny typy projektů, nezávisle na charakteru dodávaného produktu či služby. Pokud by tato oblast byla zanedbána, může to vést k nepříjemným obstrukcím jako je zvýšení časové náročnosti projektu, resp. zvýšení celkových nákladů na projekt.

2.7.1 Plánování řízení kvality

Jde o proces identifikace kvalitativních požadavků na projekt a jeho cíle a stanovení základních principů dodržování kvality. Dochází tedy k jasnému vymezení potřeb projektu v rámci řízení kvality. Plánování kvality je prováděno současně s jinými plánovacími procesy, jako je tvorba plánu požadavků na projekt a tvorba plánu rizik. Je to z toho důvodu, že je třeba zvážit různé faktory vyskytující se napříč jednotlivými plány, tak aby došlo ke vzájemné integraci.

Vstupy:

- zakládací listina projektu,
- projektové dokumenty,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení kvality,
- metriky kvality,
- úpravy dalších souvisejících projektových dokumentů,
- úpravy plánu řízení projektu.

2.7.2 Řízení kvality

Činnosti prováděné v rámci tohoto procesu do značné míry vychází ze skutečností uvedených v procesu předchozím – prakticky se jedná o převedení tohoto plánu do konkrétních aktivit, které zajistí dodržení kvalitativních požadavků v průběhu provádění projektu.

Vstupy:

- projektové dokumenty,
- plán řízení projektu,
- OPA.

Výstupy:

- zpráva o kvalitě,
- testovací a hodnotící dokumenty,
- seznam požadavků na změnu,
- úpravy plánu řízení projektu,
- úpravy dalších souvisejících projektových dokumentů.

2.7.3 Kontrola kvality

V rámci tohoto procesu jsou pozorovány a zaznamenávány výsledky provádění činností nutných pro dodržení kvality. Smyslem tohoto snažení je zajištění toho, že dodané výsledky projektu budou v souladu s požadavky zákazníka. Dochází tedy ke kontrole produktu/slужby, zda je schopna naplnit očekávání všech zúčastněných stran. Aby bylo dosaženo tohoto cíle a produkt/slужba splnila akceptační kritéria zákazníka, je vhodné provádět tento proces v průběhu celého projektu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- odsouhlasené požadavky na změny,
- seznam cílů,
- data o výkonu práce,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- výsledky měření kvality,
- ověřené cíle,
- informace o výkonu práce,
- seznam požadavků na změny,
- úpravy plánu řízení projektu,
- úpravy dalších souvisejících projektových dokumentů.

2.8 Řízení zdrojů projektu

Řízení zdrojů projektu obsahuje procesy, které slouží pro identifikaci, rezervaci a řízení zdrojů potřebných k úspěšnému dokončení projektu. Přínosem těchto procesů je zastřešení projektu z hlediska zdrojů – a to, že budou konkrétní zdroje k dispozici ve správný čas na správném místě. Je třeba rozlišovat zdroje ve smyslu pracovníků, týmových členů, lidských sil a zdroje ve smyslu věcném, jako jsou materiály nutné pro vytvoření produktu, jeho komponenty, či další zařízení nutná pro výkon činností projektu. Protože klíčovými zdroji projektu jsou jednotliví týmoví členové, otevírá se zde obrovská příležitost uplatnění měkkých dovedností projektového manažera, který by měl umět tým řídit, smelovat a být jeho srdcem.

Tato znalostní oblast je poměrně rozsáhlá, obsahuje 6 procesů, které náleží, stejně jako předchozí oblast, do tří skupin procesů (plánovací, prováděcí a monitorovací). Má i podobný charakter a stavbu procesů.

2.8.1 Plánování řízení zdrojů

V rámci procesu jsou popsány způsoby jakými odhadnout, získat, řídit a užívat věcné a lidské zdroje. Je určen způsob jakým budou získávány specifické zdroje nutné pro úspěch projektu. Je nutné počítat i s možností, že požadované zdroje nebudou v daný okamžik k dispozici, např. z důvodu jejich momentálního využívání jiným projektem s vyšší prioritou. Tyto situace je tedy nutné řešit předem a pokud se vyskytnou nečekaně, je třeba pružně zareagovat.

RACI matice

Zajímavou pomůckou pro řízení zdrojů, která je využívána pro určení zodpovědnosti je matice přiřazení zodpovědnosti RACI. Ta popisuje způsob jakým jsou jednotlivé osoby (členové týmu/společnosti) zodpovědné za konkrétní činnosti. U každé aktivity jsou členům určeny role (odpovědný, vyžadováno uzavření podpisem, konzultant, informován). RACI do jisté míry souvisí s WBS, protože může obsahovat právě činnosti, které jsou popsány hierarchickým rozkladem.

Vstupy:

- základací listina projektu,

- projektové dokumenty,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení zdrojů,
- úpravy dalších souvisejících projektových dokumentů,
- základací listina týmu.

2.8.2 Odhad množství zdrojů

Cílem tohoto procesu je určení typu a množství zdrojů nutných pro provedení projektu. Může se jednat o počet kusů materiálů, vybavení, množství surovin a dalších zdrojů, které je nutné mít k dispozici.

Vstupy:

- projektové dokumenty,
- plán řízení projektu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- požadavky na zdroje,
- seznam faktů pro odhad zdrojů,
- resource breakdown structure (analogie WBS pro zdroje),
- úpravy dalších souvisejících projektových dokumentů.

2.8.3 Získávání zdrojů

Je procesem získávání lidských zdrojů, vybavení, zařízení, materiálů a dalších zdrojů potřebných k dokončení projektu. Přínosem projektu je přiřazení zdrojů k činnostem. Potřebu zdrojů, které jsou získatelné z majetků společnosti je nutné řešit s příslušným manažerem. Zdroje, které je třeba získat z externích míst, jsou řešeny obstarávacími procesy. Projektový manažer, případně ostatní pověřeni členové týmu, by měli být schopni vyjednávat o možnostech alokace potřebných zdrojů, pokud jsou tato vyjednávání neúspěšná, může to mít negativní dopad na plán projektu, rozpočet, kvalitu a spokojenost zákazníka.

Vstupy:

- plán řízení zdrojů,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- přidělení věcných zdrojů,
- přidělení lidských zdrojů,
- kalendář zdrojů,
- seznam požadavků na změny,

- upravený plán řízení projektu,
- upravené další související projektové dokumenty,
- upravené faktory podnikového prostředí,
- upravené OPA.

2.8.4 Utváření týmu

Utváření týmu je procesem zdokonalování projektového týmu. Během tohoto procesu, který probíhá po celou dobu trvání projektu, jsou vylepšeny schopnosti, týmový duch a týmové prostředí. Důvodem existence tohoto procesu je předpoklad, že splněný cíl projektu je produktem schopného a dobře fungujícího týmu. Stěžejní roli v tomto procesu má projektový manažer, musí být schopen vytvořit, motivovat, vést a inspirovat tým k dosažení vytyčeného cíle. To do jaké míry bude tento proces úspěšný tedy do velké míry záleží na dovednostech manažera.

Fáze týmu

Dr. Bruce Tuckman představil v roce 1965 model [2], který představuje formování týmu v průběhu času. Je složen z 5 fází, které na sebe ve většině případů navazují.

- 1 **Formování** – dochází k sestavení týmu, jednotliví členové jsou si vzájemně představeni, jsou určeny role v týmu.
- 2 **Krize** – dochází k ní pokud mezi sebou mají jednotliví členové nesrovnalosti, které vedou k neschopnosti týmové spolupráce.
- 3 **Normalizace** – přichází ve chvíli, kdy k sobě členové opět najdou cestu a jsou schopní produktivní práce.
- 4 **Provádění** – tým dosahuje svého rozkvetu a to jak po stránce mezilidských vztahů, tak především po stránce výkonové. Jsou schopni samostatně řešit i pokročilé úkoly, což by bylo v předchozích fázích nemyslitelné.
- 5 **Přerušeni** – nastává typicky při dokončení práce na projektu, nebo při rozpuštění týmu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- hodnocení výkonu týmu,
- seznam požadavků na změnu,
- upravený plán řízení projektu,
- upravené další související projektové dokumenty,
- upravené faktory podnikového prostředí,
- upravené OPA.

2.8.5 Vedení týmu

Vedení týmu je proces pozorování výkonu jednotlivých členů, poskytování zpětné vazby, řešení problémů a provádění změn v týmu, tak aby došlo k optimalizaci týmového výkonu. Tento proces probíhá v průběhu celého projektu a ze strany vedoucího týmu požaduje jisté dovednosti, které souvisí s uměním vést tým.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- zpráva o vykonané práci,
- hodnocení výkonu týmu,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam požadavků na změnu,
- upravený plán řízení projektu,
- upravené další související projektové dokumenty,
- upravené faktory podnikového prostředí.

2.8.6 Kontrola zdrojů

Je procesem zjišťování, zda alokované zdroje projektu, jsou skutečně dostupné. Také se jedná o zajišťování změn v požadavcích na zdroje, tak aby byly uspokojeny požadavky týmu a došlo ke splnění naplánované práce. Současně je také nutné využité zdroje posléze uvolnit, např. pro potřeby jiných týmu. Je vhodné poznamenat, že v rámci procesu *Kontroly zdrojů* se jedná o věcné zdroje jako je vybavení, materiál, pomůcky. Personální potřeby zdrojů řeší předchozí proces Vedení týmu.

Vstupy:

- plán řízení projektu,
- projektové dokumenty,
- zpráva o vykonané práci,
- smlouvy,
- OPA.

Výstupy:

- informace o výkonu práce,
- seznam požadavků na změny,
- upravený plán řízení projektu,
- upravené další související projektové dokumenty.

2.9 Řízení komunikace v rámci projektu

Sedmá znalostní oblast se týká komunikace. Spadají sem procesy, které zajišťují komunikační stránku projektu a zúčastněných stran, řeší se zajištění komunikačních kanálů a realizují se činnosti, které zajišťují informační výměnu. Je třeba vybudovat vhodný komunikační protokol. Komunikace může probíhat např. pomocí e-mailu, sociálních médií, projektových reportů, atd. Současně je možné provádět výměnu informací na osobní úrovni např. na projektovém setkání či prezentaci. Všechny tyto aspekty by tedy měly být ujasněny a pevně stanoveny. Projektový manažer bývá často úzkým místem komunikace mezi týmem a zúčastněnými stranami, měl by tedy být obdařen jistou komunikační a sociální inteligencí, protože je pravděpodobné, že bude muset domlouvat záležitosti s osobami, které mohou mít jinou komunikační kulturu, než on sám/sama.

2.9.1 Plánování řízení komunikace

Smyslem procesu plánování řízení komunikace je vytvoření způsobu a plánu pro komunikační činnosti v rámci projektu, který zohledňuje zájmy a potřeby zúčastněných stran a potřeby projektu.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- dokumenty (dokumentace požadavků, rejstřík zúčastněných stran),
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení komunikace,
- upravený plán řízení projektu.

2.9.2 Řízení komunikace

Řízení komunikace je proces zajištění včasného sběru, tvorby, ukládání, sledování a konečného uspořádání informací o projektu. Smyslem je vytvoření dobře fungujícího toku informací mezi projektovým týmem a zúčastněnými stranami.

Vstupy:

- plán řízení projektu,
- dokumenty (protokol změn, zpráva o kvalitě, zpráva o rizicích, rejstřík zúčastněných stran),
- zpráva o výkonu prací,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- komunikace v rámci projektu,
- řízení komunikace,
- upravený plán řízení projektu,
- upravené související vstupní dokumenty,
- upravené OPA.

2.9.3 Sledování komunikace

Je procesem zajišťujícím, že informační potřeby projektu a zúčastněných stran jsou uspokojeny. Sleduje, zda dochází k plnění plánu řízení komunikace a plánu zapojení zúčastněných stran.

Vstupy:

- plán řízení projektu (plán řízení zdrojů, plán řízení komunikace, plán zapojení zúčastněných stran),
- dokumenty (registr problémů, komunikace v rámci projektu),
- data o výkonu práce,
- faktory podnikového prostředí,

- OPA.

Výstupy:

- informace o výkonu práce,
- žádosti o změnu,
- upravený plán řízení projektu,
- upravené další související projektové dokumenty.

2.10 Řízení obstarávání v rámci projektu

Tato znalostní oblast obsahuje procesy, které jsou nutné pro zajištění chodu projektu z pohledu obstarávání. Jedná se o procesy, které řeší nákup či nabytí produktů a služeb z vnějšího prostředí. Rovněž řeší smluvní záležitosti ohledně dohod, objednávek, či zajištění licencovaného softwaru. Z hlediska právního se jedná o velmi důležitou oblast, protože zanedbání některého z procesů může vést k právní žalobě, resp. k soudním pokutám a sankcím. Projektový manažer zde většinou není tím, kdo by měl poslední slovo při podpisu smlouvy, pro tuto činnost jsou ve společnosti obvykle jiné osoby s patřičným vzděláním či zkušenostmi. Ve znalostní oblasti jsou přítomny tři procesy náležící do tří skupin procesů (plánovací, realizační a monitorovací).

2.10.1 Řízení plánování obstarávání

Je procesem evidence projektových rozhodnutí o obstarávání, popisem způsobu obstarání a výběrem možných dodavatelů, kteří mohou být osloveni. V rámci procesu je rozhodnuto o tom, zda konkrétní produkty/služby budou pořízeny a také je popsán způsob jakým bude dosaženo jejich získání.

Vstupy:

- zakládací listina projektu,
- obchodní dokumenty,
- plán řízení projektu,
- projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán řízení obstarávání,
- strategie obstarávání,
- zpracované nabídky od dodavatelů,
- prohlášení o výkonu práce (ohledně obstarávání),
- kritéria rozhodování o dodavatelích,
- rozhodnutí zda produkt vyrobit či koupit,
- nezávislý odhad cen,
- požadavek na změny,
- úpravy dalších souvisejících příslušných dokumentů.

2.10.2 Vedení obstarávání

Jedná se o proces poptávání, vyhodnocení nabídek včetně výběru dodavatel a udělení zakázky. Cílem procesu je vybrat nejlepšího kandidáta a zajistit vhodné právní pozadí dohody.

Vstupy:

- plán řízení projektu,
- dokumentace požadavků,
- nabídky dodavatelů,
- faktory podnikového prostředí,
- OPA.

Výstup:

- zvolený dodavatel,
- uzavřená smlouva,
- požadavky na změny,
- úpravy plánu řízení projektu,
- úpravy dalších souvisejících příslušných dokumentů.

2.10.3 Kontrola obstarávání

Je procesem řízení vztahů s dodavateli, sledování vývoje zakázky, provádění případných změn v zakázce. V průběhu procesu je zajištěno, že dojde k uspokojení potřeb obou stran a zároveň k dodržení dohod.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- dokumentace obstarávání,
- smlouvy,
- dokumentace požadavků,
- odsouhlasené požadavky na změny,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- ukončené smlouvy o obstarávání,
- upravená dokumentace požadavků,
- upravená dokumentace obstarávání,
- požadavky na změny,
- úpravy plánu řízení projektu,
- úpravy dalších souvisejících příslušných dokumentů.

2.11 Řízení zúčastněných stran

Znalostní oblast řízení zúčastněných stran obsahuje procesy nutné pro identifikaci osob, skupin a organizací, které by mohly libovolným způsobem ovlivňovat průběh projektu, či naopak, na které by mohl působit projekt. Jedná se o souhrn aktivit, jejichž výsledkem je popis způsobů začlenění identifikovaných osob, skupin, organizací a také popis strategie způsobu jejich zapojení v projektu. Jde o důležitou část projektu, protože schopnost uspokojit potřeby zúčastněných stran často souvisí s finálním úspěchem projektu, tuto potřebu je tedy možné vnímat i jako jeden z jeho dalších cílů. Proces identifikace a analýzy těchto osob, skupin, organizací je třeba začít co nejdříve po vytvoření základací listiny projektu.

Vzhledem k charakteru tohoto faktoru, kdy se často mění např. požadavky na projekt ze strany účastníků, je vhodné zvolit některou z agilních metodik, což by mělo vést ke snížení rizika neúspěchu projektu. Nabízí se přímý kontakt projektového týmu se zúčastněnými stranami namísto komunikace, které je vhodné přímo začlenit do rozhodovacího procesu projektu.

2.11.1 Identifikace zúčastněných stran

Identifikace zúčastněných stran je prvním procesem oblasti. Dochází zde k identifikaci osob, skupin, firem, organizací, které mohou mít vliv a dopad na průběh projektu, po jejich identifikaci následuje analýza jejich zájmů, vlivů a angažovanosti na činnostech projektu, případně na jeho cíli. Typicky se toto děje na začátku projektu, v jeho iniciační fázi, po vytvoření základací listiny, je ale vhodné tento proces opakovat po každé významné změně.

Vstupy:

- základací listina,
- plán řízení projektu,
- další projektové dokumenty,
- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- rejstřík zúčastněných stran,
- požadavky na změny,
- upravený plán řízení projektu,
- upravené další související projektové dokumenty.

2.11.2 Plánování zapojení zúčastněných stran do projektu

Je proces vytvoření způsobu zapojení zúčastněných stran do projektu. Během tohoto procesu jsou zohledňovány jejich zájmy, očekávání a možný dopad na projekt. Výstupem je plán, který popisuje a shrnuje jakým způsobem jsou zapojeni účastníci a jak s nimi jednat. Přímo navazuje na předchozí proces identifikace, ze kterého vychází. Výsledný plán je však nutné měnit pokud dojde k určitým okolnostem, např. změna složení množiny zúčastněných stran nebo při změnách v organizační struktuře firmy.

Vstupy:

- základací listina projektu,
- plán řízení projektu,
- další projektové dokumenty,
- smlouvy,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- plán zapojení zúčastněných stran.

2.11.3 Řízení zapojení zúčastněných stran do projektu

Proces řídí zapojení účastníku do projektu. Řeší spolupráci s účastníky, komunikuje s nimi, a to vše za účelem minimalizace možných negativních důsledků plynoucích z jejich možného dopadu na projekt. Tímto způsobem je zvýšena šance na úspěch.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstup:

- požadavky na změnu,
- upravený plán řízení projektu a další související dokumenty.

2.12 Řízení rizik projektu

Desátá a zároveň závěrečná znalostní oblast se týká řízení rizik projektu. Tato oblast a zejména procesy identifikace a plánování rizik v ní obsažené budou popsány podrobněji, než tomu bylo v předchozích případech. Ještě před samotným rozbořem procesů je vhodné vysvětlit některé základní pojmy, které budou dále obsaženy.

Riziko

Ottů slovník naučný definuje riziko (risiko) následovně: „*risiko jest odvaha, se kterou čelí podnikatel možné ztrátě při svém podniku. V přeneseném smysle znamená ztrátu samu.*“. Toto vysvětlení pojmu je jistě korektní, nicméně neobsahuje přímou souvislost s projektovým řízením. Riziko může mít v širším pojetí několik různých definicí. Dle prof. Raise ([3], s. 28), např.: „*pravděpodobnost jakéhokoliv výsledku, odlišného od výsledku očekávaného*“, nebo: „*odchýlení skutečných a očekávaných výsledků*“. To, jak je riziko interpretováno, tedy do jisté míry závisí na tom, v jaké oblasti je tento pojem použit.

Klasifikace rizik

Rizika mohou být klasifikována ([4], s. 78 – 81), dle jejich charakteru, následovně:

- *dynamické riziko* – má příčinu ve firmě/organizaci samotné, ekonomických poměrech, nebo v nejbližším okolí firmy,
- *statické riziko* – nesouvisí s firmou, ale může mít např. původ v přírodních živlech, katastrofách, či může souviset s cíleným poškozením ze strany osob,
- *čisté riziko* – jde o případy, kdy z případné situace není možné mít prospěch a může nastat pouze negativní zkušenost,
- *spekulativní riziko* – protipól čistého rizika, typicky se jedná o rizikové obchody (akcie, burza), kdy je možné ztratit, ale i získat – tedy po výskytu události se osoba, která riziko podstoupila, může nacházet v komfortnějších poměrech, ale také může ztratit.

Aktivum

Pod pojmem aktivum se rozumí veškeré statky, které mohou být ohroženy vystavením se hrozbě. Tyto statky mohou být charakteru hmotného i nehmotného. V případě *hmotných aktiv* se jedná zejména o fyzický majetek (např. budovy, výrobky, stroje). V kontextu IT oblasti

může být za tento typ aktiva považován třeba finanční majetek firmy, který může být v ohrožení. Do *nehmotných aktiv* se zahrnuje například software, goodwill firmy, licence.

Protiopatření

Jedná se o veškeré kroky, které firma může podniknout, a které povedou k redukcí rizika či k redukcí dopadu hrozby v případě jejího uskutečnění. Tyto postupy je však nutné pečlivě, vhodně a adekvátně naplánovat. Je typické, že na protiopatření jsou vynakládány finanční prostředky, jejichž výše by pochopitelně neměla překročit možnou způsobenou škodu. Zde se nabízí otázka, jak případnou finanční ztrátu nejlépe vyjádřit tak, aby odpovídala skutečnosti. Pravidlem tedy je, že výše nákladů na protiopatření by měla zohledňovat nejen faktický dopad hrozby na aktiva, ale také pravděpodobnost jejího výskytu (čím nižší, tím by i náklady měly být nižší).

Hrozba

Hrozbou se rozumí cokoli, co může mít negativní dopad na aktiva. Zpravidla je charakterizována zejména mírou dopadu, což může být numerické vyjádření velikosti vzniklých škod.

Zranitelnost

Zranitelnost vyjadřuje skutečnost, že aktivum může přijít ke ztrátě způsobené v důsledku aplikace hrozby. Úroveň zranitelnosti se klasifikuje následujícími dvěma faktory ([7], s. 53):

- *citlivost* – náchylnost aktiva být poškozeno danou hrozbou a
- *kritičnost* – důležitost aktiva pro analyzovaný subjekt.

Klíč: 1 = nejnižší stupeň vyspělosti, 5 = nejvyšší stupeň vyspělosti				
Znalostní oblast	Strojírenství a stavebnictví	Telekomunikace	Informační systémy	Výroba špičkových technologií
Rozsah	3.52	3.45	3.25	3.37
Čas	3.55	3.41	3.03	3.50
Náklady	3.74	3.22	3.20	3.97
Kvalita	2.91	3.22	2.88	3.26
Lidské zdroje	3.18	3.20	2.93	3.18
Komunikace	3.53	3.53	3.21	3.48
Rizika	2.93	2.87	2.75	2.76
Obstarávání	3.33	3.01	2.91	3.33

Obr. 2: Vyspělost procesů oblastí v jednotlivých oborech (převzato z [5], s. 433)

Oblast řízení rizik je jednou z nejkritičtějších a nejdůležitějších ve sféře řízení projektu. Je to způsobeno tím, že v podstatě každá činnost či proces (tyto dva pojmy jsou míněny v obecném smyslu, nikoliv v kontextu PMI) s sebou přináší možný neúspěch, vůči kterému je nutné nějakým způsobem bojovat.

W. Ibbs a Y. H. Kwak vypracovali studii ([5], s. 432), ve které je zanalyzována situace 38 společností, které byly vybrány napříč různými obory. Průzkum probíhal tak, že respondenti odpovídali na otázky, které měly vypovědět o tom, jak moc kvalitně jsou řízeny procesy v jednotlivých znalostních oblastech. Výsledky lze vidět na Obr. 2. Je zajímavé a poměrně alarmující, že oblast rizik má téměř ve všech oborech nejnižší hodnocení. V případě

informačních technologií je situace ještě smutnější, protože řízení rizik si odneslo nejnižší hodnocení, tedy nejnižší zodpovědnost při řízení.

2.12.1 Plánování řízení rizik

Jedná se o první proces této oblasti. Je zde stanoveno jakým způsobem bude prováděno řízení rizik v projektu. Je nutné, aby výstup, kterým je plán řízení rizik, byl k dispozici v co nejranější fázi projektu a mohl tak být nasazen do provozu.

Vstupy:

- *Zakládací listina projektu.*
- *Plán řízení projektu* – konkrétně by se mělo jednat o všechny dílčí plány ze všech oblastí. Je to z toho důvodu, aby mohla být stanovena příslušná rizika z nich plynoucí.
- *Další projektové dokumenty* – zejména rejstřík zúčastněných stran, ze kterého by měly být patrné charaktery a záměry jednotlivých účastníků, což jsou faktory, které rovněž zasahují do tvorby rizikového plánu.
- *Faktory podnikového prostředí,*
- *OPA* – zde se může jednat např. o různé znalostní báze ohledně postupů podniku v oblasti rizik či dokumentaci rolí projektu a přidružených zodpovědností.

Užité nástroje

Při plánování rizik je nutné charakterizovat jakým způsobem bude probíhat jejich řízení. Je užitečné, aby proběhla schůzka, na které mohou být přítomni zástupci zúčastněných stran, projektový manažer, členové týmu, případně další odborníci, kteří se vyznají v problematice řešeného projektu. Tato schůzka by měla proběhnout co nejdříve od zahájení projektu.

Výstupy:

Výstupem tohoto procesu je plán řízení rizik. Jedná se o jednu z dílčích částí plánu projektu. Níže následuje seznam dokumentů, které mohou být v plánu řízení rizik zahrnuty.

- *Riziková strategie* shrnuje celkový přístup k řízení rizik projektu.
- *Metodologie* poskytující souhrn specifických pomůcek a nástrojů pro vypořádání se s riziky.
- *Role* a k nim přiřazené *zodpovědnosti* za činnosti, které jsou vykonávány v rámci řízení rizik.
- *Popis financování* aktivit spojených s řízením rizik.
- *Načasování* procesů spojených s řízením rizik v rámci provádění projektového řízení.
- *Kategorie rizik.* Pro identifikaci rizik (bude specifikována v následující kapitole) je možné zvolit přístup rozkladu podobně jako v jiných procesech, např. ve Vytváření WBS. Jedná se o risk breakdown structure (RBS), která obsahuje hierarchické rozčlenění rizik do kategorií, které jsou dále rozděleny do dalších subkategorií. Toto členění probíhá tak dlouho, dokud není z poslední úrovně možné snadno identifikovat související rizika. Stromová struktura RBS je patrná z obrázku Obr. 3.

RBS LEVEL 0	RBS LEVEL 1	RBS LEVEL 2
0. ALL SOURCES OF PROJECT RISK	1. TECHNICAL RISK	1.1 Scope definition
		1.2 Requirements definition
		1.3 Estimates, assumptions, and constraints
		1.4 Technical processes
		1.5 Technology
		1.6 Technical interfaces
		Etc.
	2. MANAGEMENT RISK	2.1 Project management
		2.2 Program/portfolio management
		2.3 Operations management
		2.4 Organization
		2.5 Resourcing
		2.6 Communication
		Etc.
	3. COMMERCIAL RISK	3.1 Contractual terms and conditions
		3.2 Internal procurement
		3.3 Suppliers and vendors
		3.4 Subcontracts
		3.5 Client/customer stability
		3.6 Partnerships and joint ventures
		Etc.
	4. EXTERNAL RISK	4.1 Legislation
		4.2 Exchange rates
		4.3 Site/facilities
		4.4 Environmental/weather
		4.5 Competition
		4.6 Regulatory
		Etc.

Obr. 3: Risk breakdown structure (převzato z [1], s. 442)

- *Rizika plynoucí se zapojení zúčastněných stran* – každý z účastníků může mít jiné zájmy, které jsou mnohdy protichůdné se zájmy jiných účastníků – tento fakt může být zdrojem rizika, které by nemělo být opomenuto. Je nutné správně určit do kterých oblastí budou tato rizika působit.
- *Stanovení pravděpodobnosti výskytu rizika a jeho dopadu* – pravděpodobnost rizika je nutno určit na základě charakteru projektu, vlivu účastníků a dalších okolností. Granularita škály dopadu často vyplývá z plánu řízení rizik, kde je definována, může to být například následující:
 - nízký – minimální dopad na průběh,
 - střední – závažný dopad na průběh,
 - vysoký – likvidující dopad na celý projekt.

Tato data bývají shrnuta do přehledných tabulek, ze kterých lze snadno vyčíst jaké má příslušné riziko pravděpodobnost výskytu a také stanovení dopadu na určité klíčové aspekty projektu jako je výše nákladů či doba dodání.

- *Mapa rizik* (Obr. 4.) – jedná se o grafické vyjádření skutečností vyjádřených v předchozím bodě, tedy míry pravděpodobnosti výskytu rizika a dopadu. Míra dopadu může být vynášena např. na svislou osu a pravděpodobnost na osu horizontální.

Vzhledem k faktu, že pokud má riziko spekulativní charakter, je možné u něj definovat i možný pozitivní dopad. I tato skutečnost bude promítnuta do mapy rizik. Rizika, která mají negativní dopad budou posuzována jako hrozby, rizika s příznivým dopadem na projekt (jejich důsledkem může být např. snížení nákladů či urychlení prací) jsou chápány jako příležitosti. Mapa rizik může být rozdělena do kvadrantů, které charakterizují riziko.

- *Definice formátu zprávy*, která bude zpracována na výstupu procesů z oblasti řízení rizik.
- *Popis sledování* stanovuje jakým způsobem budou zaznamenávány činnosti řízení rizik.

		Threats					Opportunities						
Probability	Very High 0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09	0.05	Very High 0.90	Probability
	High 0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04	High 0.70	
	Medium 0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03	Medium 0.50	
	Low 0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02	Low 0.30	
	Very Low 0.10	0.01	0.01	0.02	0.04	0.08	0.08	0.04	0.02	0.01	0.01	Very Low 0.10	
		Very Low 0.05	Low 0.10	Moderate 0.20	High 0.40	Very High 0.80	Very High 0.80	High 0.40	Moderate 0.20	Low 0.10	Very Low 0.05		
Negative Impact						Positive Impact							

Obr. 4: Mapa rizik s příležitostmi a hrozbami (převzato z [1], s. 408)

2.12.2 Identifikace rizik

Je procesem pochopení zdrojů rizik a jejich rozlišení v rámci projektových činností. Je důležité, aby tento proces identifikace provádělo pokud možno co nejvíce relevantních osob vzhledem k projektu. Mezi typické účastníky, kteří se na této činnosti podílí patří podobně jako v předchozím procesu projektový manažer, projektový tým, zástupci zúčastněných stran, zástupci firmy (např. vedoucí finančního oddělení, oddělení zdrojů, atd.) a další osoby.

Vstupy:

- *Plán řízení projektu*, konkrétně se může se jednat o následující části:
 - *Plán řízení požadavků* (výstup procesu Plánování řízení rozsahu), ze zajišťování požadavků mohou plynout příslušná rizika, která je nutno identifikovat.
 - *Časový plán projektu* (výstup procesu Řízení časového plánu projektu), pro identifikace časových rizik.
 - *Plán řízení nákladů* (výstup procesu Plánování řízení nákladů), pro identifikaci rizik plynoucích z této oblasti.

- *Plán řízení kvality* (výstup procesu Plánování řízení kvality), pro identifikaci rizik plynoucích z této oblasti.
- *Plán řízení zdrojů* (výstup procesu Plánování řízení zdrojů), pro identifikaci rizik plynoucích z této oblasti.
- *Plán řízení rizik* (výstup procesu Plánování řízení rizik), který může být využitý pro určení zodpovědnosti za rizika.
- *Plán řízení rozsahu* (výstup procesu Plánování řízení rozsahu), součástí tohoto plánu může být WBS, která může posloužit pro identifikaci rizik.
- *Základní časové schéma* (výstup procesu Tvorba časového rozvrhu), časový harmonogram specifikuje milníky práce – je třeba stanovit rizika, která mohou dosažení těchto milníků překazit.
- *Odhad rozpočtu* (výstup procesu Stanovení rozpočtu), pro stanovení rizik, která mohou mít za důsledek překročení rozpočtu.
- *Další projektové dokumenty* – zejména odhady cen, odhad dob trvání, dokumentace požadavků, rejstřík získaných poznatků, rejstřík požadavků a rejstřík zúčastněných stran.
- *Smlouvy*.
- *Dokumentace obstarávání* (výstup procesu Kontrola obstarávání), požadavky na různé zdroje mohou být zdroji rizik, které je nutno vzít v potaz.
- *Faktory podnikového prostředí*.
- *OPA* – zde se může jednat např. o soubory s aktuálními informacemi o dění v projektu.

Užité nástroje

Je vhodné posouzení situace ze strany odborníka na danou problematiku, případně dalších osob, které mají zkušenosti s podobnými projekty. Jejich znalosti mohou být nedoceníitelné a stěží získatelné z jiných zdrojů. Pro další získávání informací na základě kterých mohou být identifikována rizika související s projektem patří například brainstorming. Brainstorming je metoda, kdy se skupina osob snaží na první pohled nahodile vymýšlet nápady, které mohou být relevantní. Je nutné, aby v této skupině lidí byli přítomni všichni, kdo mají k příslušnému tématu co říct. Velmi důležitá je také role moderátora diskuze, který by měl být schopen řídit tuto debatu tak, aby se nějakým způsobem nezvrhla a vedla k odpovídajícím výsledkům. Dalším přístupem mohou být rozhovory, ve kterých je vyzpovídána osoba disponující příslušnou znalostní bází.

Pro analýzu dat jsou používány různorodé techniky. Jde například o *root cause analysis (RCA)*, která si klade za cíl zjistit zdroje příčin problému a funguje na principu eliminace těchto zdrojů – tedy pokud nejsou příčiny problému, nemůže existovat ani problém samotný. Další známou technikou je *SWOT analýza*. Cílem SWOT analýzy je identifikace silných, slabých stránek, příležitostí a hrozeb. Nejdříve by měla proběhnout určení silných a slabých stránek organizace projektu. Poté dojde na fázi určení hrozeb, které plynou ze slabých stránek a příležitostí plynoucích ze stránek silných. Více informací o těchto postupech je uvedeno v kapitole 3.1.

Výstupy:

Výstupem procesu identifikace rizik je několik dokumentů. Primárním výstupem, který plyne ze samotné podstaty procesu je:

- *Registr rizik* – obsahuje seznam konkrétních rizik a příslušných informací o každém z nich. Jak takový registr rizik může vypadat, je možné spatřit na obrázku Obr. 5.

Číslo	Stupnice	Riziko	Popis	Kategorie	Prvotní příčina	Spouštěče	Potenciální reakce	Vlastník rizika	Pravděpodobnost	Dopad	Stav
R44	1										
R21	2										
R7	3										

Obr. 5: Příklad registru rizik. (převzato ze [5], s. 447)

Registr rizik by měl obsahovat všechny podstatné informace ohledně zacházení s rizikem. Jednotlivé položky mají přiřazeny unikátní identifikátor (číslo), které je možné využít pro odkazování. Každé riziko by mělo být přiřazeno zodpovědnou osobu, která toto riziko řídí a reaguje na něj (vlastník rizika).

- *Zpráva o rizicích* – dalším dokumentem, který by měl být výstupem tohoto procesu je zpráva o rizicích. Jedná se o přehled rizik projektu a měla by obsahovat identifikaci zdrojů rizik a souhrnné informace o rizicích rozpoznaných v projektu.
- *Úprava dalších projektových dokumentů*, které byly zasaženy změnami provedenými v rámci tohoto procesu (např. rejstřík získaných poznatků).

2.12.3 Provedení kvalitativní analýzy rizik

Smyslem procesu kvalitativní analýzy je určení závažnosti rizika na základě posouzení jejího dopadu a pravděpodobnosti. Prakticky jsou tedy jednotlivá identifikovaná rizika seřazená dle závažnosti.

Vstupy:

- plán řízení projektu (plán řízení rizik),
- další projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- upravené projektové dokumenty (např. registr rizik).

2.12.4 Provedení kvantitativní analýzy rizik

Proces kvantitativní analýza provádí číselné vyhodnocení dopadů rizik. Výsledkem je získání dalších informací o dopadu rizika na projekt, které mohou přijít vhod při rozhodování o reakcích na rizika.

Tento typ analýzy není nutný provádět pro všechny projekty, zpravidla je vykonávána u projektů většího rozsahu, za kterými stojí větší instituce či firma. Mnohokrát je také spojena s předchozím typem analýzy, kvalitativní, do jednoho celku ([8], s. 451).

Je zde využíváno několik zajímavých metodik pro stanovení dopadu rizik na projekty. Jedná se například o analýzu pomocí rozhodovacích stromů. Z jednotlivých uzlů vychází větve (hrany), přičemž součet jejich ohodnocení musí dávat hodnotu jedna, které značí s jakou

pravděpodobností proběhne určitá událost. Na základě těchto faktů je pak možné se rozhodovat v klíčových situacích. Dalším prvkem, který je používán je simulace, kde je aplikována analýza Monte Carlo.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- upravené projektové dokumenty (zpráva o rizicích).

2.12.5 Plánování reakcí na rizika

Jedná se o proces rozhodování o reakcích na identifikovaná rizika. Jde o vytváření těchto reakcí, jejich plánování a plánování strategií pro zvládání rizik. Smyslem procesu je vytvořit adekvátní reakce na rizika projektu. Rozlišuje se několik základních reakcí v závislosti na charakteru rizika, které jsou shrnuty v tabulce Tab.1.

	Vysoká pravděpodobnost	Nízká pravděpodobnost
Vysoký dopad	Vyhnutí se riziku, redukce	Pojištění
Nízký dopad	Retence a redukce	Retence

Tab. 1: Doporučené metody pro obecný přístup k riziku. (převzato z [4], s. 86)

Z tabulky je vidět, že pro rizika vyznačující se vysokým dopadem i pravděpodobností se vyplatí vyvinout protipatření či riziko nějakým jiným způsobem redukovat. Pro rizika s vysokým dopadem a nízkou pravděpodobností je vhodné volit pojištění (stejně jako v ostatních oblastech života). U rizika s nízkým dopadem a vysokou pravděpodobností je nutné rozhodnout, zda je dopad tak drastický, aby se vyplatilo podstoupit protipatření. Protože v těchto případech, kdy jsou následky snesitelné, je možné riziko vypustit a podstoupit. Samozřejmě je ale nutné i toto zdánlivě nepodstatné riziko bedlivě sledovat a to z toho důvodu, že například v pozdějších fázích může mít znatelnější dopad. Přístup který může být uplatněn pro rizika, která jsou málo pravděpodobná a která představují malý dopad na projekt, je retence. Je to z toho důvodu, že takových rizik je nepřehledné množství a nemá význam řešit každé jedno z nich, protože náklady vynaložené na agendu tohoto typu rizik by jistě přesáhly své přínosy.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam požadavků na změnu,
- upravený plán řízení projektu,
- upravené další dokumenty.

2.12.6 Uskutečnění reakcí na rizika

Je procesem realizace reakcí na rizika dle ustanovení vzniklých v předchozím procesu. Smyslem procesu je zajistit, že dohodnuté postupy jsou skutečně provedeny a dodržují se.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- faktory podnikového prostředí,
- OPA.

Výstupy:

- seznam požadavků na změnu,
- upravení další související dokumenty.

2.12.7 Sledování rizik

Sledování rizik je procesem kontroly provedených reakcí na rizika, sledování identifikovaných rizik a vyhodnocováním řízení rizik v rámci projektu. Aby došlo k co největšímu zajištění, že projekt proběhne dle očekávání, je nutné sledovat práci na projektu a pokud se objeví nový zdroj rizik, musí být zahrnut do registru rizik a musí vůči němu vzniknout správná protiopatření.

Vstupy:

- plán řízení projektu,
- další projektové dokumenty,
- data o vykonané práci,
- zprávy o vykonané práci.

Výstupy:

- informace o vykonané práci,
- seznam požadavků na změnu,
- plán řízení projektu,
- upravení další dokumenty,
- upravené OPA.

3 Specifikace požadavků a návrh

Vzhledem k charakteru zadání práce, které se zaměřuje na plánování a identifikaci rizik, je při specifikaci požadavků nutné zohledňovat zejména výstupy těchto dvou procesů. Rozbor obou procesů byl poměrně důkladně proveden v předchozích kapitolách tohoto textu. V rámci této kapitoly budou diskutovány funkční požadavky na aplikaci pro podporu managementu rizik, které vycházejí z nastudované metodiky. Oba procesy na sebe navazují a do jisté míry se prolínají. Velice srozumitelně jsou výstupy z této oblasti shrnuty na stránkách PMI [8].

Primárním výstupem procesu Plánování řízení rizik je komplexní sumarizací přístupů k řízení rizik daného projektu. Jedním z charakteristických výstupů je stanovení šablony pro rejstřík rizik. Každé odvětví, každý projekt je specifický svými charakteristikami, které je nutné monitorovat při řízení rizik. Kromě typických atributů rejstříku rizik, kterými mohou být např. identifikační číslo rizika, jeho vlastník, kategorie, existuje mnoho dalších atributů, které mohou být do rejstříku zahrnuty. Výsledkem je tedy finální podoba šablony, která bude používána později při identifikaci rizika. Dalším výstupem je stanovení hodnot a kategorizace pravděpodobnosti a dopadu rizika na projekt. Tento aspekt je zachycen v podobě matice pravděpodobností a dopadu. Příklad realizace této matice je na obrázku Obr. 6.

Impact	Probability			
		High	Medium	Low
	High	H	H	M
	Medium	H	M	L
	Low	M	L	L

Obr. 6: Matice pravděpodobností a dopadu. (převzato z [8])

Poslední záležitostí, kterou je nutno stanovit, je určení rolí a zodpovědností v rámci aktivit plánu řízení rizik. V procesu Identifikace rizik, který je de facto stěžejním pro celou znalostní oblast, dochází k identifikaci rizik. Primárním a vlastně i jediným cílem je tedy, na základě znalostí pozadí projektu, vytvořit registr rizik, jehož schéma je definované v předchozím procesu.

Dá se očekávat, že aplikace najde využití zejména v kancelářích, kde jsou IT projekty nejen realizovány, ale i řízeny. Z tohoto důvodu je vhodné ji navrhnout na webovém platformě, tedy jako webovou aplikaci.

3.1 Analýza požadavků pro identifikaci rizik

Kapitola si klade za cíl popsat způsob směru dat a informací a jejich následné vyhodnocení pro účely projektu. Této problematice se již věnovala kapitola 2.12.2 Identifikace rizik, nyní je vhodné si zmíněné nástroje rozebrat do větší hloubky a popsat jejich klíčové vlastnosti, cíle a vlastně i samotný způsob práce s těmito metodami. Tuto záležitost je možné rozdělit do dvou samostatných celků, kterým obsahově odpovídají následující dvě části.

3.1.1 Shromažďování dat

Prvotním úkolem projektového týmu je mít k dispozici co největší množství fakt, informací, skutečností, zpráv a poznatků týkajících se projektu a jeho zúčastněných stran. Je pochopitelné a snadno ověřitelné, že pokud bude projektový manažer (či jiná osoba zodpovědná za řízení rizik – pro usnadnění bude v dalším textu použit pouze termín projektový manažer) mít k dispozici nesprávné, neúplné nebo nekorektní informace, může jen stěží správně identifikovat rizika a to i za použití ideálních metod při následné datové analýze.

Brainstorming

Je metodou, která by měla vygenerovat značné množství nápadů a idejí. Obvykle se dělí na dvě na sebe chronologicky navazující části – *získávání nápadů* a jejich *vyhodnocování*. Klíčové je, aby probíhala za přítomnosti tzv. *facilitátora* – jedná se o vybranou osobu, která by měla dohlížet na správný průběh debaty, měl by být schopen zajistit, aby se skupina příliš neodchylovala od původního tématu a zaměření. Facilitátor by také měl být do jisté míry vůdčí osobností, nebo alespoň by měl být způsobilý sjednat pořádek a např. zajistit, aby se všichni členové brainstormingového týmu dostali ke slovu a nedocházelo pouze k výměnám názorů mezi např. dvěma nebo třemi jedinci.

V případě identifikace rizik je možné postupovat tak, že jsou postupně systematicky procházeny jednotlivé kategorie rizik (Obr. 3). K těmto událostem mohou být přizváni i externí odborníci z příslušných oblastí, kteří nemusejí být součástí projektového týmu, avšak jejich znalosti mohou být přínosem pro tento proces.

Kontrolní seznamy

Kontrolní seznam je pomůcka sloužící pro identifikaci rizik projektu. Jedná se o seznam úkonů, položek či činů, které je nutné zvážit při realizaci projektu. Firmy často mívají vypracované své interní seznamy, které různě modifikují a upravují na základě momentálních potřeb aktuálně řešeného projektu. Pokud se např. jedná o malou začínající firmu, je možné dohledat na internetu obecné podoby těchto rejstříků. V praxi tento seznam může být realizován jako seznam otázek, které je třeba zodpovědět a na základě získaných informací je možné přistoupit k určení rizik z nich vyplývajících. Otázky přítomné v seznamech mohou být kategorizovány do několika celků – rizika plynoucí z obchodního dopadu, procesní rizika, rizika spojená se zákazníkem, technická rizika, rizika vývojového prostředí, atd. V rámci života projektu by mělo docházet k pravidelným aktualizacím těchto seznamů, tak aby reflektovaly současné informace o prostředí projektu.

Rozhovory

Rozhovory jsou způsobem získávání informací skupinou či jednotlivcem na straně tazatele, který/která pokládá/jí otázky skupině či jednotlivci. Dotazovaným může být člen týmu, externí konzultant, zástupce zúčastněné strany, či kdokoliv, jehož názor může mít při rozhodování o zdroji rizik určitou váhu. Pokud se v kolektivu vyskytuje méně extrovertní jedinec, kterému dělá problém projevit se před ostatními a sdílet s nimi svůj názor, může být velmi přínosné zrealizovat s ním rozhovor o samotě, mimo dosah a doslech ostatních.

Všechny tyto uvedené techniky jsou využitelné a v praxi kombinovatelné, pro účely aplikace je výhodné vybrat *kontrolní seznamy*. Důvodem je, že softwarová podpora pro tuto metodu pravděpodobně bude v praxi užitečnější, než např. pro rozhovory.

3.1.2 Analýza dat

Analýza dat je krokem, který by měl pomoci k určení relevantních rizik reálně ohrožujících projekt. Je možné, že projektový tým shromáždí velké množství dat a informací, které je třeba vhodným způsobem analyzovat a rozřadit do takové podoby, která bude sloužit jako vstup dalším procesům této oblasti (konkrétně by mělo jít o Kvantitativní a Kvalitativní analýzy). K tomu, aby bylo dosaženo tohoto stavu, je možné použít několik technik a postupů, v následujících řádcích budou postupně představeny jednotlivé možnosti, které je v praxi možné kombinovat.

Analýza kořenových příčin (angl. *Root cause analysis*)

Jak již plyne z názvu, jde o postup, při jehož aplikaci je třeba nejdříve identifikovat základní problémy, které mohou vzniknout při realizaci projektu. Smysl této metody tkví v eliminaci těchto potenciálních problémů. Tímto způsobem by tedy mělo být zajištěno, že zamýšlený scénář proběhne s úspěchem.

Analýza předpokladů a omezení

Každý projekt počítá s nějakými předpoklady, které by měly být splněny pro jeho úspěšné dokončení. Nesplnění těchto předpokladů lze samozřejmě chápat jako hrozbu, se kterou by mělo být odpovídajícím způsobem naloženo. Jako omezení jsou chápány faktory, které nějakým způsobem omezují práci týmu, často se jedná o omezení v rovině času, zdrojů, financí, atd.

SWOT analýza

SWOT analýza se zabývá silnými a slabými stránkami, příležitostmi a hrozbami (z angl. *Strengths* – silné stránky, *Weaknesses* – slabé stránky, *Opportunities* – příležitosti a *Threats* – hrozby) a slouží jako základní nástroj strategického plánování ([5] s. 141).

Jedná se o jednu z obecně nejznámějších business analýz. Tento typ analýzy má velmi široké uplatnění a lze ji uplatnit vlastně kdekoliv, kde je potřeba něco zhodnotit nebo rozhodnout. Někdy je také dělena do dvou podcelků – tzv. vnitřní a vnější faktory. Do *vnitřních faktorů* se řadí silné a slabé stránky. Jde tedy o interní aspekty hodnoceného celku, které jsou hodnoceny s odhlédnutím od vnějšího prostředí. *Vnější faktory* jsou příležitosti a hrozby působící na systém – je pro ně charakteristické, že působí nezávisle na vnitřním prostředí subjektu (typicky se jedná např. o zhodnocení konkurenčního prostředí firmy). Po identifikaci faktorů a jejich rozdělení do zmíněných čtyř faktorů je na řadě určení strategie, kterou management využije při svém dalším rozhodování – rozdělení strategií je v tabulce Tab.2.

	Silné stránky (S)	Slabé stránky (W)
Příležitosti (O)	Strategie SO	Strategie WO
Hrozby (T)	Strategie ST	Strategie WT

Tab.2: Znázornění strategií plynoucích z analýzy SWOT (zdroj: vlastní)

Existují čtyři směry, kterými se může management vydat:

- *strategie SO* – spočívá ve využití příležitostí silnými stránkami,

- *strategie WO* – spočívá ve využití příležitosti a následné minimalizaci slabých stránky,
- *strategie ST* – spočívá ve využití silných stránek k potlačení hrozeb,
- *strategie WT* – spočívá v nalezení způsobu jak minimalizovat hrozby při současném uvažování slabých stránek.

SWOT analýzu ovšem lze využít i v rámci identifikace rizik. To, jakým způsobem naloží projektový manažer s konkrétními informacemi, je pouze na jeho uvážení.

Analýza dokumentů

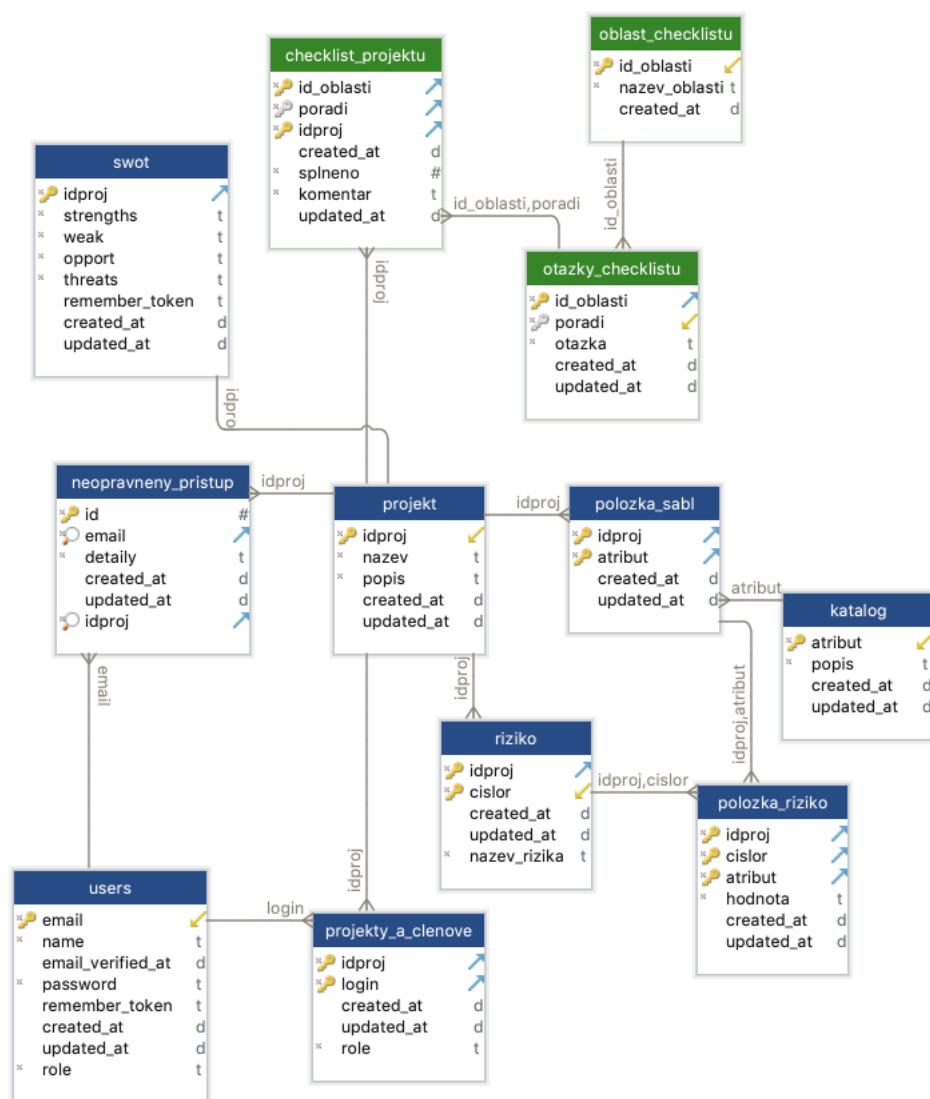
Poslední zmíněnou analýzou je analýza dokumentů, která se soustředí na nastudování a přezkoumání projektových dokumentů a rizik z nich plynoucích. Rizika mohou plynout například z nejasné formulace či nesrovnalostech napříč dokumentu. Všechny tyto indicie mohou nasvědčovat vzniku hrozby, kterou je třeba mít pod kontrolou.

Pro potřeby návrhu jsem si zvolil implementaci podpory pro řízení SWOT analýzy do výsledné aplikace.

3.2 Návrh schématu databáze

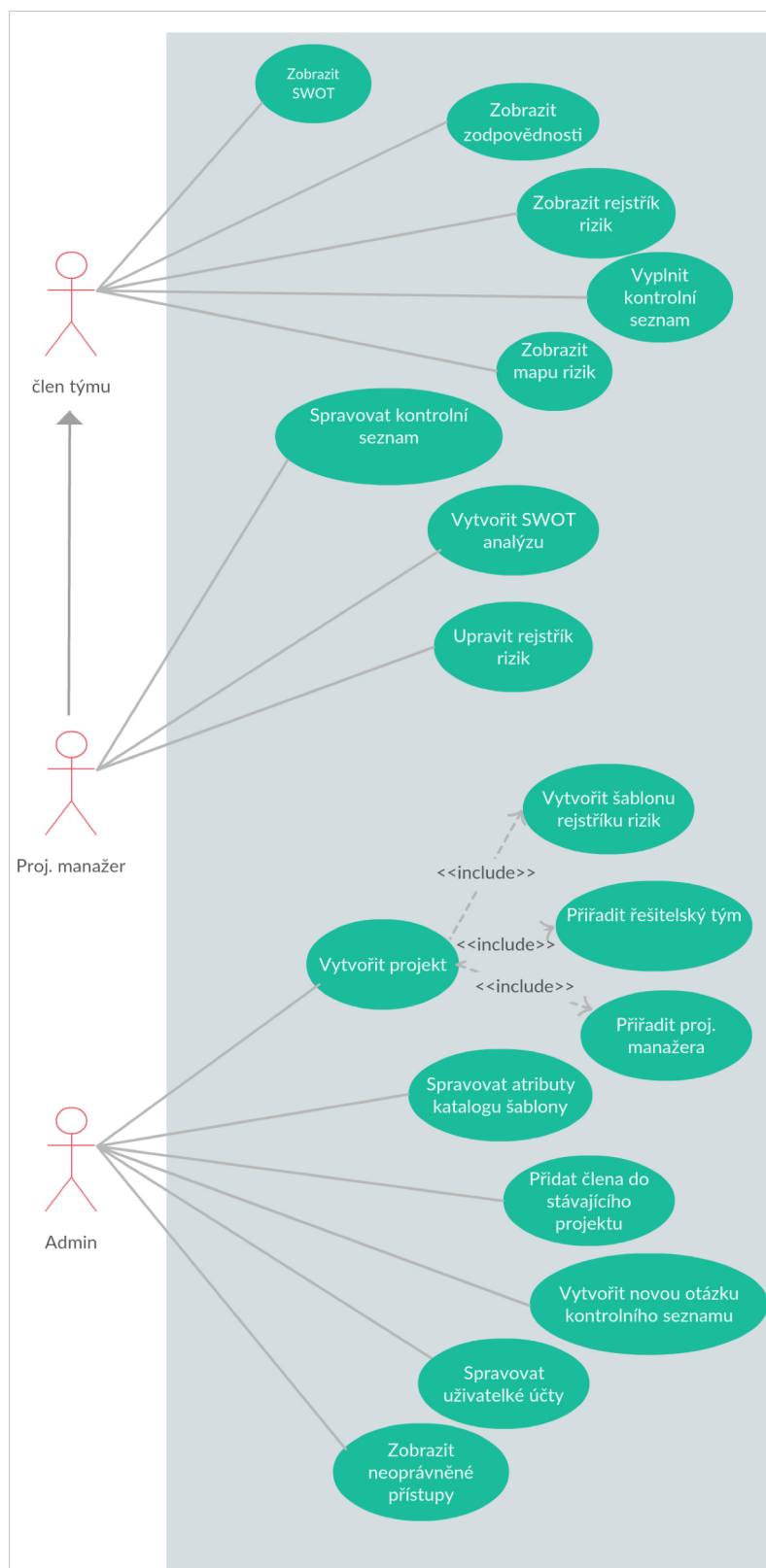
Dle specifikace požadavků byl vytvořen návrh databáze. Vizualizace schématu databáze je na Obr. 7. Jedná se celkem o 12 tabulek. Níže následuje jednoduchý popis co která tabulka znázorňuje. Ostatní informace, například primární/cizí klíč, je možné vyčíst ze schématu.

- *projekt* – představuje záznam o projektu,
- *swot* – představuje SWOT analýzu konkrétního projektu,
- *neopraveny_pristup* – tabulka, ve které jsou zachovány události neoprávněných přístupů do databáze,
- *projekty_a_clenove* – vazební tabulka, vyjadřuje který uživatel je zapsán ve kterém z projektů,
- *riziko* – obsahuje výčet všech rizik všech projektů,
- *polozka_riziko* – jde o vazební tabulku, popisuje hodnoty atributů rizik,
- *polozka_sabl* – popisuje šablony rejstříků rizik projektů,
- *katalog* – obsahuje všechny přiřaditelné atributy rejstříků rizik,
- *checklist_projektu* – obsahuje informace o kontrolních seznamech jednotlivých projektů (jde o vazební tabulku),
- *otazky_checklistu* – je to banka s otázkami, které je možné přiřadit kontrolnímu seznamu projektu,
- *oblast_checklistu* – obsahuje výčet oblastí otázek, kterými jsou naplněny kontrolní seznamy,
- *users* – tabulka uživatelů systému.



Obr. 7: Schéma databáze (zdroj: vlastní)

3.3 Diagram případů užití a identifikace aktérů



Obr. 8: Diagram případů užití (zdroj: vlastní)

Na obrázku Obr. 8 je vyobrazen diagram případů užití, ve kterém vystupují tři aktéři. Regulérní *člen týmu* má kromě přihlášení a odhlášení možnost **zobrazit SWOT analýzu** projektu, **zobrazit rejstřík rizik**. Dále má možnost **vyplnit kontrolní seznam**, který byl zadán manažerem projektu a **zobrazit matici pravděpodobností a dopadů**.

Paleta případů užití *projektového manažera* je rozšířením možností člena týmu zejména o administrativní úkony. Obsahuje **vytvoření kontrolního seznamu**, který mohou upravovat členové týmu. Je zodpovědný za obsah **SWOT analýzy**, který **vytváří**. Má možnost **stanovit zodpovědnosti**. Poslední rolí je *administrátor*, který má na starosti celkovou **agendu uživatelských účtů**, vytváří a spravuje je. Dále má možnost **vytváření projektů**, včetně **skladby řešitelského týmu** a obsahu **šablony rejstříku rizik**. Pokud je tedy zapotřebí, vytvoří v databázi nového uživatele a předá mu přihlašovací údaje. Může také vytvářet **nový obsah pro kontrolní seznamy** a **zobrazit neoprávněné přístupy**, tedy pokud se uživatel neoprávněně pokouší provádět akce s databází, ke kterým nemá pravomoci. Následuje strukturovaný popis vybraných případů užití:

Případ užití: Upravit rejstřík rizik
Účastníci: <ul style="list-style-type: none"> projektový manažer
Vstupní podmínky: <ol style="list-style-type: none"> 1. Uživatel (člen týmu) má založený účet a je přihlášen. 2. Je vytvořen projekt. 3. Uživatel je přiřazen k projektu. 4. Šablona pro rejstřík rizik je vytvořena. 5. Rejstřík rizik projektu je založen.
Tok událostí: <ol style="list-style-type: none"> 1. Uživatel zvolí projekt, ke kterému se rejstřík rizik vztahuje. 2. Systém vybere informace, které zobrazí jako seznam rizik projektu. 3. Uživatel vybere konkrétní riziko, které chce upravit. 4. Systém zobrazí aktuální hodnoty přiřazených atributů. 5. Uživatel upraví příslušné hodnoty atributů. 6. Uživatel potvrdí změny. 7. Systém uloží nové hodnoty.
Následné podmínky: <ol style="list-style-type: none"> 1. Ostatním uživatelům je k dispozici nová verze rejstříku rizik.

Případ užití: Zobrazit mapu rizik
Účastníci: <ul style="list-style-type: none"> člen týmu
Vstupní podmínky: <ol style="list-style-type: none"> 1. Uživatel (člen týmu) má založený účet a je přihlášen. 2. Je vytvořen projekt. 3. Uživatel je přiřazen k projektu. 4. Šablona pro rejstřík rizik je vytvořena. 5. Šablona rejstříku rizik obsahuje atributy Dopad a Pravděpodobnost.

6. Rejstřík rizik projektu je založen. 7. Rejstřík rizik projektu je naplněn riziky (alespoň jedním).
Tok událostí: 1. Uživatel zvolí projekt, pro který chce zobrazit mapu rizik. 2. Systém na základě hodnot atributů jednotlivých rizik vytvoří mapu rizik.
Následné podmínky: 1. Uživateli je vizuálně zobrazena matice vybraného projektu vytvořená na základě aktuálních dat.

Případ užití: Stanovit zodpovědnosti
Účastníci: <ul style="list-style-type: none"> proj. manažer
Vstupní podmínky: 1. Uživatel (proj. manažer) má založený účet a je přihlášen. 2. Je vytvořen projekt. 3. Uživatel je přiřazen k tomuto projektu jako proj. manažer. 4. Je vytvořena šablona registru rizik. 5. Šablonu registru rizik obsahuje atribut Risk-owner. 6. Registr rizik obsahuje nejméně jedno riziko. 7. Projektu jsou přiřazeni členové týmu.
Tok událostí: 1. Uživatel zvolí projekt, pro který chce stanovit zodpovědnosti za riziko. 2. Systém vybere z příslušných tabulek informace, které zobrazí jako seznam rizik projektu. 3. Uživatel vybere konkrétní riziko, které chce upravit. 4. Systém zobrazí atributy rizika. 5. Uživatel u atributu Risk-owner zvolí zodpovědnou osobu.
Následné podmínky: 1. Činnosti je přiřazen konkrétní člen týmu, který za ni nese zodpovědnost

Případ užití: Vytvořit novou otázku kontrolního seznamu
Účastníci: <ul style="list-style-type: none"> administrátor
Vstupní podmínky: 1. Uživatel má založený účet a je přihlášen. 2. Je vytvořen projekt. 3. Uživatel má roli administrátora.

<p>Tok událostí:</p> <ol style="list-style-type: none"> 1. Uživatel (administrátor) zvolí možnost přidat novou otázku do banky kontrolních seznamů. 2. Uživatel (administrátor) vytvoří znění otázky a přiřadí oblast. 3. Systém uloží data.
<p>Následné podmínky:</p> <ol style="list-style-type: none"> 1. Nově vytvořená otázka je k dispozici i ostatním proj. manažerům (resp. ostatním projektům).

Případ užití: Spravovat kontrolní seznam
<p>Účastníci:</p> <ul style="list-style-type: none"> • proj. manažer
<p>Vstupní podmínky:</p> <ol style="list-style-type: none"> 1. Uživatel má založený účet a je přihlášen. 2. Je vytvořen projekt. 3. Uživatel je přiřazen k tomuto projektu jako proj. manažer.
<p>Tok událostí:</p> <ol style="list-style-type: none"> 1. Uživatel zvolí projekt pro úpravu (přidání otázky). 2. Systém zobrazí aktuální podobu kontrolního seznamu a dále možné otázky, které mohou být přiřazeny. 3. Uživatel (proj. manažer) vybere otázky, které chce přidat do projektu. 4. Systém uloží nové informace.
<p>Následné podmínky:</p> <ol style="list-style-type: none"> 1. Obsah kontrolního seznamu vybraného projektu je obohacen o nové otázky.

4 Implementace

Tato kapitola se věnuje implementaci. Je zde popsána realizace webové aplikace pomocí aplikačního rámce. Ačkoliv jsem doposud nepřišel do styku s žádným aplikačním rámcem, rozhodl jsem se, že bude vhodné některý z nich pro realizaci využít. Níže je uvedeno porovnání jednotlivých rámců včetně podrobnějšího popisu mnou vybraného.

4.1 Výběr aplikačního rámce

Aktuálním trendem v oblasti vývoje webových aplikací je užívání aplikačního rámce (*framework*) při vývoji webu. Použití rámce přináší v podstatě samá pozitiva – práce s ním je rychlejší, přehlednější, snadnější. Mnoho z nich má také svou implementaci zajišťuje jistou úroveň bezpečnosti výsledné aplikace vůči útokům. U mnoha z nich také uživatel (programátor) nemá na výběr a musí při vývoji použít návrhový vzor MVC, což by mělo vést k čitelnějšímu kódu a ve výsledku např. i k pohodlnější práci v týmu.

4.1.1 Porovnání PHP aplikačních rámců

Vzhledem k faktu, že nemám s žádným z rámců zkušenost, byl jsem nucen zjistit si o nich patřičné informace, zanalyzovat je a rozhodnout se, který z nich použiji při vývoji. Postupoval jsem tak, že jsem si našel žebříček oblíbenosti rámců (na Obr. 9) a o prvních čtyřech jsem si zjišťoval hlubší informace, které jsou níže shrnuty.

Laravel

Jedná se o relativně mladý projekt z roku 2012. Má nejvyšší procentuální míru zastoupení mezi PHP rámci (srovnání je na Obr. 9). Vysoká oblíbenost souvisí s velkým množstvím výukových materiálů na internetu. Je pro něj charakteristické, že obsahuje velký počet zabudovaných funkcionalit, které usnadňují práci např. při tvorbě autentizace, práci s databází či směrováním.

Code Igniter (CI)

Rámec PHP původně z roku 2006 [12]. Je datově velmi nenáročný (instalační balík má pouze 2MB), podporuje architekturu MVC. Snaží se poskytnout jednoduché prostředí, které obsahuje vše potřebné pro vývoj aplikace. Negativem CI je absence šablonovacího systému pro grafické uživatelské rozhraní a také nutnost instalace případných rozšíření – což pochopitelně souvisí s malou velikostí celkových souborů rámce.

Symfony

Další rozšířenou nadstavbou PHP je Symfony. Jedná se o robustní, výkonný aplikační rámec. Podobně jako předchozí dva je velmi dobře zdokumentován. Bývá mu vytýkán poněkud těžkopádný přístup k databázi. Oproti Code Igniteru ovšem obsahuje i šablonovací systém stejně jako Laravel (ačkoliv nutno podotknout, že Laravel a CI využívají různé šabl. systémy).

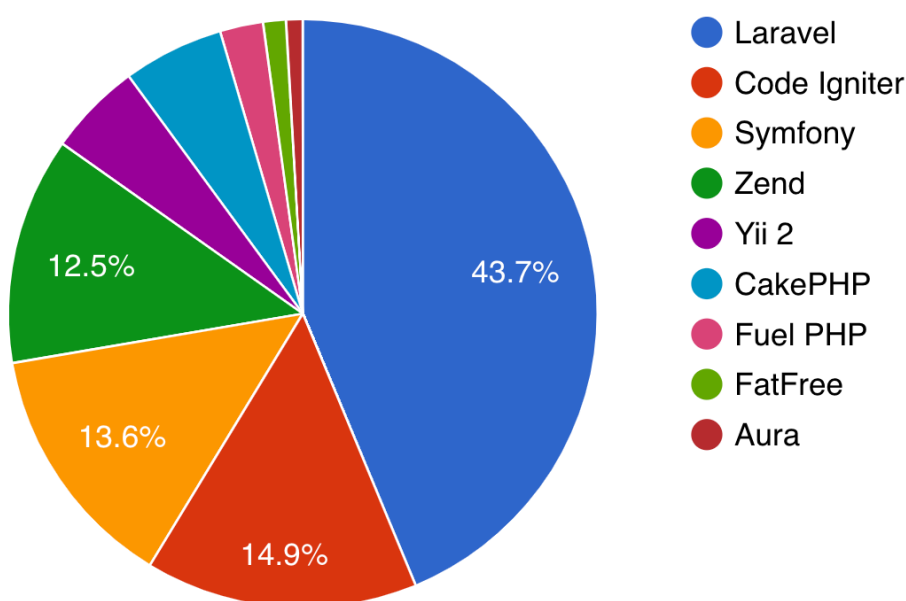
Zend

Zend je uzpůsoben spíše pro velké, komplexní projekty. Má poměrně malou uživatelskou základnu. Rovněž z pohledu začínajícího programátora se jeví jako ne příliš dobrá volba, protože neposkytuje takové množství výukových materiálů jako např. Symfony nebo Laravel a jistě se o aplikaci, kterou chci realizovat, nedá říci, že je to komplexní projekt.

4.1.2 Popis Laravelu

Pro implementaci byl vybrán rámec Laravel [9], výběr padl zrovna na něj z několika důvodů. Prvním z nich je perfektní dokumentace, mnohdy v podobě video návodů [10], jejichž autoři jsou skvělí lektoři. Tato výuková videa velmi usnadňují proces učení a jsou produkována přímo autory zdrojového kódu Laravelu. Dále je možné dohledat velké množství tutoriálů, které vysvětlují a popisují základní koncepce jazyka. V neposlední řadě hraje svou roli také uživatelská základna, která je široká – obecně není problém dohledat řešení eventuálních problémů, které se vyskytnou při implementaci. To je důsledkem velké rozšířenosti Laravelu ve světě.

Z grafu na Obr. 9, který vychází z aktuálních dat z roku 2019, je patrné, že Laravel je aktuálně jednoznačně nejpoužívanějším aplikačním rámcem na světě.



Obr. 9: Užití aplikačních rámců PHP v projektech (zdroj [11])

4.1.2.1 Základní vlastnosti Laravelu

Přestože mnoho PHP aplikačních rámců sdílí společné vlastnosti, je vhodné si vyjmenovat a přiblížit některé základní aspekty a komponenty Laravelu.

MVC

Návrhový vzor MVC (zkratka pro *model-view-controller*) nabádá programátora k rozdělení aplikace do tří skupin komponent a byl vyvinut v rámci programovacího prostředí jazyka Smalltalk-80 [13]. Jedná se o následující tři komponenty: *Model* – logický model aplikace, v případě Laravelu, který používá tzv. ORM model (objektově relační modelování) se vlastně jedná o jednotlivé tabulky uložené v perzistentní databázi. *View (pohled)* – je zodpovědný za UI a zprostředkovává zobrazení GUI uživateli, v případě webových aplikací tedy jde o značky jazyka HTML. *Controller (řadič)* – poslední komponenta stojí za logikou aplikace a řídí funkci obou předchozích komponent, fakticky provádí operace s modelem, které přicházejí jako požadavky od uživatele a následně zobrazí výsledek na výstup v podobě změny pohledu. Benefitem tohoto přístupu je nezávislost komponent mezi sebou ve smyslu modifikací.

Artisan

Artisan je pomocný nástroj přístupný přes terminál. Obsahuje sadu příkazů, které usnadňují práci při vývoji. Jedná se zejména o práci s migracemi (bude představeno dále), jejich vytváření, exekuci, tvorbu modelů, tvorbu řadičů. Například příkaz: `php artisan make:controller FlightController --resource` vytvoří v příslušném adresáři soubor představující řadič, který již bude obsahovat základní třídní strukturu a také prázdné funkce pro realizace základních operací.

Blade

Jedná se o šablonovací systém vyvinutý pro účely rámce. Šablonovací systém obecně umožňuje efektivní práci s GUI v PHP. Pro současné aplikace je typické, že mají některé prvky opakující se napříč celým webem. Blade umožňuje tyto sekce importovat pomocí odkazů mezi sebou. Není tedy nutné stále se opakující části stránek kopírovat do příslušných souborů, ale stačí se na ně odkázat pomocí klíčového slova `extends`.

ORM Eloquent a migrace

Laravel využívá pro účely databáze návrhový vzor *ActiveRecord*. Práce s databázemi probíhá tak, že pro každou tabulku si vývojář vytvoří *model*, který zajišťuje spolupráci s příslušnou databázovou tabulkou.

```
public function store(Request $request)
{
    $kniha = new Kniha;

    $kniha->autor = $request->autor;

    $kniha->save();
}
```

Obr. 10: Ukázka práce s ORM Eloquent (zdroj: vlastní)

Na obrázku Obr. 10 je uveden příklad práce s databází. Je zde nejprve vytvořena nová instance třídy *Kniha*, poté je z požadavku, který mohl být odeslán například prostřednictvím formuláře, extrahována položka *autor* a její hodnota je nastavena jako hodnota atributu *autor* nově vytvořené instance. Celý objekt je poté uložen jako nový záznam do příslušné tabulky pomocí metody `save()`. Obdobným způsobem je možné zpracovat všechny standardní databázové příkazy.

Zajímavým prvkem jsou *kolekce*, které lze chápat jako rozšíření datového typu pole, jsou používány mj. pro manipulaci s daty. Například pro výběr položek [14] (analogie příkazu `SELECT`) je postup takový, že jsou na prvky kolekce (celý obsah tabulky) postupně pomocí šipkové notace (`->`) aplikovány jednotlivé metody. Programátor tedy nemá důvod používat jazyk *SQL*, protože všechny příkazy lze vytvořit pomocí existujících metod.

Migrate [15] jsou soubory, které jsou využívány pro vytváření a modifikaci návrhu databáze. Jedná se o soubory, které obsahují definici migrační třídy dvěma základními metodami `up()` a `down()`. Metoda `down()` je volána při rollbacku databáze a měla by tedy logicky obsahovat příkazy, které zvrátí to, co metoda `up()` vytvoří nebo nastaví. Pro vytváření migrací se používá prostředí *artisan*, příkaz `make:migrate` vytvoří migraci, příkaz `migrate` ji provede.

Bezpečnost

Obecně platí, že aplikační rámce poskytují vyšší úroveň zabezpečení než čistá forma jazyka. Laravel není výjimkou. Eloquent účinně zabraňuje SQL injekcím, protože používá výše zmíněný způsob tvorby SQL příkazů. Dalším možným typem útoku je tzv. *Cross-site request forgery*, jedná se o podvržení požadavku mezi stránkami. Ve zkratce se jedná o to, že útočník vytvoří a odešle požadavek na server. Tento požadavek je odeslán prostřednictvím účtu autorizovaného uživatele a server tedy nebude mít pochybnosti o jeho pravosti a zdárně ho vykoná. Laravel tento problém řeší tak, že ke každé seanci (angl. session) přiřadí tzv. *CSRF „token“*. Z pohledu vývojáře tak stačí, aby ke každému formuláři připojil pole CSRF „token“. Takto odeslaný formulář bude na úrovni tzv. *middleware* (vrstva zpracovávající požadavky) zkontrolován a jeho token bude porovnán proti tokenu uloženému v session.

Další kapitolou je autentizace. Laravel poskytuje zabudovanou uživatelskou podporu pro její vytváření a správu. Pomocí příkazu `artisan make:auth` lze vytvořit všechny nezbytné pohledy a směrování (tzv. *routování*). Hesla jsou před uložením do databáze zahashována. Nikde tedy není k dispozici jejich původní forma. Např. pro ověření uvnitř aplikace, zda je uživatel úspěšně přihlášen, slouží metoda `check()`. Není tedy nutné ověřovat tuto skutečnost manuálně přístupem do databáze.

4.2 Popis implementace grafického rozhraní

Implementace tedy byla realizována na webové platformě za použití aplikačního rámce Laravel. Grafické rozhraní bylo implementováno za použití webových technologií. Úplný základ tvoří značkovací jazyk HTML. Vzhled jeho prvků byl ovlivněn CSS frameworkem Bulma [16], který poskytuje příjemný, čistý a přehledný vzhled. Grafické uživatelské rozhraní (GUI) aplikací běžících pod Laravelem je generováno šablonovacím systémem Blade, jehož princip byl nastíněn v předchozích kapitolách.

V rámci Laravelu jsou všechny pohledy (obrazovky) uloženy ve složce *resources/views*. Tento adresář zpravidla obsahuje základní rozložení stránky (např. navigační menu), tzv. master layout, který sdílí všechny „synovské“ pohledy, do kterých je vložen, ale může obsahovat další složky, ve kterých mohou být uloženy opět další složky nebo pohledy. Každý takový soubor má koncovku **.blade.php*. Obsahově jde o klasický soubor s HTML značkami, který však může obsahovat různé řídicí direktivy převzaté z PHP, tyto se do zdrojového kódu zapisují pomocí zavináčové notace viditelné na obrázku Obr. 11.

```
<body>
    @if (count($zaznam) === 1)
        Je k dispozici jeden záznam
    @elseif (count($zaznam) > 1)
        Je k dispozici více záznamů
    @else
        Žádné záznamy k dispozici
    @endif
</body>
```

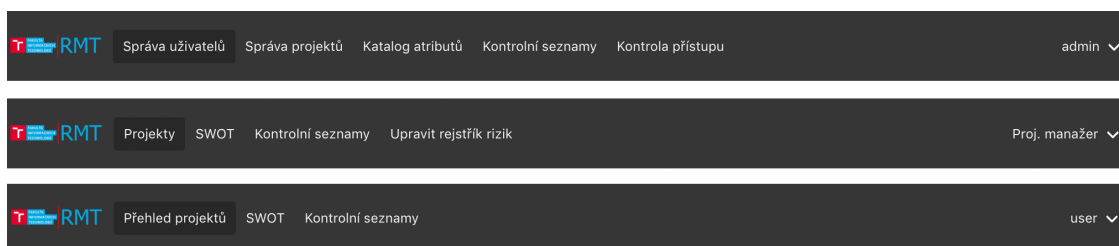
Obr. 11: Ukázka užití zavináčové notace v Blade šabloně. (zdroj: vlastní)

Často je nutné zacházet s obsahem proměnných předaných jako parametr v příslušném řadiči. K obsahu proměnné je možné přistoupit pomocí dvojitého složeného závorky: `{{ $zaznam }}`.

V následujících podkapitolách je popsán vztah mezi jednotlivými pohledy a také některá zajímavá řešení, ke kterým jsem přistoupil v rámci GUI.

4.2.1 Základní rozložení obrazovky

Pod označením základní rozložení obrazovky se rozumí výše zmíněný master layout. V dalším textu se budu držet anglického termínu, neboť dle mého názoru lépe vystihuje podstatu. Master layout se tedy nachází v kořenovém adresáři *views*. Je složen v podstatě pouze z navigační lišty a zápatí stránky. V systému figurují celkem tři aktéři, kterým náleží tři odlišné podoby menu, které reflektují jejich pravomoci. Základním kritériem, které jsem si stanovil při návrhu byla snadná dostupnost jednotlivých funkcionalit aplikace, ideálně na jeden klik. Konkrétní vzhled všech variant je k dispozici na Obr. 12.



Obr. 12: Srovnání navigačních lišt jednotlivých rolí (typů účtů) (zdroj: vlastní)

4.2.2 Obrazovky případů užití

Master layout je tedy šablonou, která se vyskytuje ve všech obrazovkách aplikace (kromě přihlašovacího rozhraní). Do této hlavní šablony je vložen pomocí klíčového slova `@extends` zbylý obsah určitého pohledu. V praxi to vypadá tak, jak je nastíněno na Obr. 13.

```
@extends('welcome')
@section('content')
    {{--

        Obsah

    --}}
@endsection
```

Obr. 13: Ukázka aplikace šablonovacího systému (zdroj: vlastní)

Master layout aplikace se nachází v souboru *welcome.blade.php*. Tento obsahuje ve svém těle sekci s názvem `content`, která je vyznačena klíčovým slovem `@yield('content')`. Na začátku každého pohledu je takto naimportován master layout následovaný příslušným obsahem.

Typický pohled obsahuje formulářové prvky, které slouží pro interakci uživatele se systémem. Všechny obrazovky využívají stylování pomocí CSS poskytované rámcem *Bulma*. Bulma obsahuje ve svých zdrojových kódech *třídy*, které lze aplikovat na prvky a modifikovat tak jejich vzhled. Většina tříd je použitelná na elementy `<div>` a ``. Veškerý obsah každého pohledu je vložen do kořenového elementu `<div>`, kterému je přidělena třída `box`. Tato třída vytvoří kolem svého obsahu stínované ohraničení a celý obsah je tímto způsobem vykreslen do vycentrovaného kontejneru. Jelikož je `<div>` blokový

element, bylo třeba vyřešit, aby se všechny obsah nezobrazoval pod sebe, což by vedlo k nepraktickému designu, kdy by byl zaplněn pouze úzký pruh uprostřed obrazovky. K tomu jsem využil responzivní třídu `columns`, která umožňuje rozdělit element na sloupce, které mohou být naplněny rozličným obsahem. Šířka jednotlivých sloupců je definována dalšími třídami jako jsou např. `is-one-third`, takto popsany `<div>` bude mít šířku třetiny svého nadřazeného elementu, `is-one-fifth`, mající pětinu, atd. Vybrané snímky obrazovek budou přiloženy v následujících kapitolách.

4.3 Popis implementace případů užití

V této podkapitole se budu věnovat popisu implementace aplikační logiky. Jak již bylo zmíněno, při vývoji byly dodržovány principy architektury MVC, která se vyznačuje tím, že odděluje data, logiku aplikace a GUI. V Laravelu je typické, že pro každý model existuje odpovídající řadič, který obstarává logiku operací prováděných s daty. Další text tedy bude rozdělen do sekcí, kde každá bude popisovat účel a operace jednoho z řadičů.

4.3.1 Modul ChecklistController

Jak již název napovídá, jde o řadič, který řeší a zajišťuje akce spojené s problematikou kontrolních seznamů. Třída sestává z vyššího množství metod, než je tomu v případě ostatních řadičů. Je to dáno tím, že agenda kontrolních seznamů je poměrně rozsáhlá – jedná se o tři tabulky, které je možné modifikovat, utvářet a přidávat nový obsah. Řadič obsahuje celkem sedm funkcí, poslední dvě jsou dostupné pouze pro administrátorské účty (role `admin`).

index()

Metoda vracející úvodní obrazovku pro správu kontrolních seznamů. Smyslem je tedy získání kolekce projektů, které jsou uživateli přiřazeny. Tato informace je získána z tabulky `checkboxlist_projektu`. V procesu tvorby projektu (resp. v příslušné metodě) je zahrnuto i vytvoření záznamu pro daný projekt v této tabulce.

show()

Funkce, která zobrazí kontrolní seznam projektu, který uživatel zvolil. Navazuje na funkci `index()`. Výsledkem je tabulky s otázkami rozřazenými do jednotlivých oblastí. Důležitou funkcionalitou je možnost vytvářet si své vlastní (uživatelské) poznámky v podobě komentářů a možnost odškrtnutí položky jako „splněno“.

update()

Metoda fakticky provádějící úpravu v databázi. Uživatel má možnost uložit si komentář a splnění otázky do tabulky `checkboxlist_projektu`. Úprava záznamů v tabulce se děje v rámci této funkce. Potřebné informace jsou získány z HTTP požadavku.

insert()

Pokud uživatel (role `projamanazer`) uváží, že obsah aktuálního kontrolního seznamu není dostačující, může do něj přidat další položky. Tato akce je zprostředkována funkcí `insert()`. Prvním krokem je získání kolekce všech otázek z databáze (jde o spojení tabulek `oblast_checkboxlistu` a `otazky_checkboxlistu`). Druhým krokem je získání obsahu kontrolního seznamu aktuálně upravovaného projektu – rovněž v podobě kolekce. Následně je implementován rozdíl těchto dvou množin (pro odstranění průniku je použita zabudovaná funkce `forget()`), který je poté (vedle, aktuálního seznamu) předán do pohledu. Uživatel tedy vidí, které položky aktuální seznam obsahuje a je schopen výběrem přidat nový obsah.

store()

Metoda vytvoří novou instanci třídy `checklistProjektu`, z HTTP požadavku získá data, která mu přiřadí a uloží jej do databáze.

admin_insert() a admin_store()

Dvojice funkcí pro administraci kontrolních seznamů. První z nich naplní pohled přehledem všech otázek všech oblastí a vytvoří jednoduchý formulář pro přidání nové otázky. Následně jsou tato data uložena do databáze a připravena pro přidání do seznamů.

4.3.2 Modul KatalogController

KatalogController obstarává případy užití týkající se katalogu atributů rejstříku rizik. Jedná se o poměrně jednoduchou záležitost. Samotná třída obsahuje tři metody. První z nich, `index()`, slouží pro zobrazení obsahu katalogu a je volána při kliknutí na odpovídající odkaz z navigační lišty. Druhá metoda, `insert()`, je zodpovědná za vložení nové položky do databáze a opětovné zobrazení aktualizovaného obsahu databáze. Metoda `destroy()` slouží ke smazání záznamu z katalogu. Je vhodné zmínit, že má-li být položka smazána, je nutné, aby nebyla přiřazena k žádnému z projektů. V opačném případě bude na tuto skutečnost uživatel upozorněn dialogovým oknem. Teprve až po smazání projektu, který má tento atribut uveden jako součást své šablony, bude operace úspěšná. Uživatel bude opět notifikován dialogovým oknem. Na Obr. 14 je ukázka pohledu souvisejícího s katalogem.

Název	Popis	Smazat
Dopad	míra dopadu výskytu na projekt	<input type="button" value="x"/>
Pravděpodobnost	míra pravděpodobnosti výskytu rizika	<input type="button" value="x"/>
Risk owner	Osoba zodpovědná za riziko	<input type="button" value="x"/>
Výše škod	finanční vyjádření dopadu na projekt	<input type="button" value="x"/>
Zdroj rizika	oblast odkud riziko pochází	<input type="button" value="x"/>
Indikátor rizika	elementy nasvědčující výskytu rizika	<input type="button" value="x"/>

Atribut

Popis

Obr. 14: Obrazovka katalogu atributů rejstříků rizik (zdroj: vlastní)

4.3.3 Modul PolozkySablonyController

Vztah, který popisuje, které atributy z katalogů náleží kterému projektu, je zachycen v databázi tabulkou `polozka_sabl`. Jedná se o tabulku, která zachycuje vztah M:N, tedy tzv. vazební tabulka. Řadič zasílá data do obrazovky detailu projektu. Detail projektu, který s těmito funkcemi souvisí je na Obr. 15. Nutno podotknout, že pro vytvoření této obrazovky, resp. pro zprostředkování akcí je využito metod více řadičů.

store()

Funkce pro přiřazení nové položky do šablony rejstříku rizik projektu. Na začátku metody proběhne ověření zda je prováděna adminem a následně je vytvořena nová instance modelu `PolozkySablony`. Poté je v cyklu také naplněna tabulka `polozka_riziko` iniciačními hodnotami atributů (pro každé riziko) a uživatel je zpětně přesměrován na detail projektu.

destroy()

Opačná operace k předchozí. Jde o smazání položky z šablony. Na úrovni databáze již není třeba explicitně provádět žádnou operaci. Příslušné související záznamy jsou odstraněny automaticky akcí `ON DELETE`.

Detail projektu Aloe Drama vol. 2 Smazat projekt

Lorem ipsum dolor sit amet. Etiam ligula pede, sagittis quis, interdum ultricies, scelerisque eu. Integer pellentesque quam vel velit.

řešitelský tým:

Tomáš Vítasek	✖
Dušan Valecký	✖
Petr Novák	✖

Přidat člena:

Franta Vomáčka ▼

Přidat

manažer projektu: Petr ProjM

Změnit:

Tomáš ProjM ▼

Změnit

šablona rejstříku rizik:

Cena škody a opravy	✖
Dopad	
Kategorie	✖
Pravděpodobnost	
Risk owner	✖
Spouštěč	✖

Přidat:

Protiopatření ▼

Přidat atribut

Obr. 15: Detail projektu (zdroj: vlastní)

4.3.4 Modul ProjektController

Jde o řadič projektů. Zprostředkovává akce spojené s vytvářením, úpravami a mazáním projektů z databáze. Paleta metod, které modul poskytuje je poměrně široká.

index()

Zjišťuje projekty, ve kterých je právě přihlášený uživatel zainteresován. Do pohledu tedy vrací kolekci projektů, ty jsou zobrazeny jako výčet a tvoří v zásadě základní rozhraní uživatelů s rolí `user` a `projmanazer`. Uživatelé je po kliknutí na libovolný projekt zobrazen detail projektu.

create()

Možnost vytvářet projekty je umožněna uživatelé s administrátorskými právy. Na tomto místě je vhodné zmínit, že jednou ze základních funkcionalit, kterou by měla aplikace nabízet je schopnost zobrazení zmíněné mapy rizik. U každého z rizik je z jeho atributů *dopad* a *pravděpodobnost* vypočtena celková hodnota rizika, která je promítnuta do mapy. Je tedy nutné, aby byly u každého rizika (resp. v šabloně každého z projektů) k dispozici tyto dvě

informace. Z tohoto důvodu není možné při vytváření vynechat atributy dopad a pravděpodobnost. Tato záležitost je ošetřena na úrovni kódu.

Všechna data (výřez obrazovky s formulářem pro tvorbu projektu je na Obr. 16) jsou získána z HTTP požadavku, jehož podobu je možné vidět na Obr. 17. V rámci této funkce jsou tedy vloženy nové záznamy hned do několika databázových tabulek. Kromě tabulky projektů se jedná i o vazební tabulku `projekty_a_clenove`, kde projektu s právě vygenerovaným `id_proj` je přiřazen `login` člena týmu. Analogicky probíhá tento scénář v tabulce `polozka_sabl`, kde je `id_proj` přiřazen atribut z katalogu.

Vytvořit nový

Název projektu

Název projektu

Popis projektu

Název projektu

Projektový manažer

Tomáš ProjM

Řešitelský tým

Tomáš Vitasek	<input type="checkbox"/>
Dušan Valecký	<input type="checkbox"/>
Franta Vomáčka	<input type="checkbox"/>
Petr Novák	<input type="checkbox"/>
Jan Doseděl	<input type="checkbox"/>

Šablona rejstříku rizik

Cena škody a opravy	<input type="checkbox"/>
Dopad	<input checked="" type="checkbox"/>
Kategorie	<input type="checkbox"/>
Pravděpodobnost	<input checked="" type="checkbox"/>
Protiopatření	<input type="checkbox"/>
Risk owner	<input type="checkbox"/>
Spouštěč	<input type="checkbox"/>
Výše škod	<input type="checkbox"/>
Zdroj rizika	<input type="checkbox"/>

Přidat Resetovat

Obr. 16: Tvorba nového projektu (zdroj: vlastní)

Za zmínku rovněž stojí, že k záznamům v tabulce je přístupováno objektovým způsobem, obdobně jako na Obr. 10, nejsou zde tedy prováděny žádné SQL příkazy.

destroy()

Funkce pro smazání veškerých záznamech o projektu z databáze. Prostřednictvím identifikačního čísla projektu je nalezen odpovídající záznam v tabulce projektů, který je odstraněn. Díky nastavené akci `ON DELETE` s parametrem `CASCADE` je zajištěno, že dojde ke smazání opravdu všech souvisejících záznamů. Tato operace je poměrně kritická z důvodů nezvratnosti případných následků, je tedy důležité ověřit, že je prováděna skutečně z administrátorského účtu - což je samozřejmě ošetřeno.

4.3.5 Modul RejstrikController

Jedná se o řadič rejstříku rizik. Zodpovídá a provádí všechny operace, které tvoří a upravují jeho obsah. Jedná se vlastně o hlavní funkcionalitu aplikace a je tedy poměrně obsáhlý jak z hlediska množství obsažených metod, tak i počtu obrazovek, se kterými interaguje. Následuje výčet a popis metod modulu.

index()

Jednoduchá funkce pro výpis projektů. Funkce je vyvolána po kliknutí na odkaz z navigační lišty. Uživatelé jsou samozřejmě zobrazeni pouze projekty, ve kterých je zapsán jako člen řešitelského týmu, ani změnou URL nelze dosáhnout jiného výsledku. Pokud se uživatel pokusí zobrazit projekt, ke kterému nemá přístup je tato akce včetně detailů zaznamenána do tabulky `neopraveny_pristup`.

```
{
  "_token": "EgoxZbUK1sEVmJh1I4ZaolKhppTzNoKm76Qhjdix",
  "_method": "PATCH",
  "nazev": "Informační systém pro SOU",
  "popis": "V rámci projektu se řeší problematika IS pro střední odborná učiliště, jedná se o středně velký projekt",
  "role": "tom@hynek.cz",
  "clenove": [
    "jan@vitas.cz",
    "dusan@valec.cz"
  ],
  "result": [
    "Dopad",
    "Pravděpodobnost",
    "Výše škod",
    "Zdroj rizika",
    "Indikátor rizika"
  ]
}
```

Obr. 17: Obsah HTTP požadavku odpovídající obsahu formuláře na Obr. 16. (zdroj: vlastní)

show()

Funkce zodpovědná za zobrazení rizik projektu, který uživatel zvolil v předchozím kroku. V pohledu, který vrací má uživatel možnost kliknout na odkaz, který jej navede na formulář pro vytvoření nového rizika k danému projektu.

update()

Po zvolení projektu a zvolení rizika, jehož parametry chce uživatel upravit, je zobrazen formulář pro modifikaci rizika. Pole formuláře obsahují položky z šablony pro rejstřík vybraného projektu včetně aktuálních hodnot. Obrazovka, která souvisí s touto akcí je na Obr. 18.

Úprava rejstříku rizik projektu

Vyberte projekt:

- Scholastic
- Aloe Drama
- Aloe Drama vol. 2
- LA IT Point
- Informační systém pro SOU

Výpis rizik:

Číslo	Název
1	Výpadek serverů
2	Změna vedení
3	Nedostatek prostředků
4	Změna termínu
5	Výměna prostředníka

[Vytvořit nové](#)

Podrobnosti rizika č.2

Dopad

3

Pravděpodobnost

1

Risk owner

dusan@valec.cz

Zdroj rizika

management

Upravit

Obr. 18: Úpravy rejstříku rizik (zdroj: vlastní)

Uživatel upraví pole dle svého uvážení a potvrdí svou volbu tlačítkem, poté je mu zobrazeno notifikační pole, které potvrzuje, že skutečně došlo k uložení nových hodnot do databáze.

store()

Metoda, která je zavolána při kliknutí na tlačítko *Upravit* (na Obr. 18). Dochází zde ke zpracování požadavku. Jedná se o cyklický průchod tělem požadavku a sekvenční úpravu jednotlivých položek. Pomocí zabudované funkce `except()` jsou odfiltrovány položky požadavku zajišťující režii – jde o `_token`, tedy prvek ochrany před CSRF útokem a `_method`, tímto způsobem je mírně urychleno provedení těchto operací.

create()

Metoda pro vytvoření nového rizika. Podobně jako při úpravě hodnot rizika jsou z databáze získány informace o šabloně rejstříku rizik. V podobě formuláře jsou poté vykreslena prázdná pole s těmito atributy. Implicitní hodnota u atributů dopadu a pravděpodobnosti je zde nastavena na hodnotu 3.

save_risk()

Požadavek, který byl vytvořen funkcí `create()`, resp. potvrzením formuláře tlačítkem, je zpracován touto funkcí. Jedná se o podobný přístup jako u metody `store()`.

4.3.6 Modul RizikaController

Oblasti působení modulů *RizikaController* a *RejstrikController* se do jisté míry protínají. Oba řadiče pracují s podobnými daty, avšak provádějí jiné akce. Smyslem tohoto řadiče je zejména zajištění práce a všech úkonů nutných pro sestavení mapy rizik. Z pohledu využitelnosti aktéry systému se jedná o modul, ke kterému mají přístup uživatelé s rolí `user` i `projmanazer`. Jedná se vlastně o interpretaci rejstříku rizik ve vizuální a grafické podobě. Vizuální podoba je dodána tabulkou a grafická podoba je demonstrována zmíněnou mapou rizik. Tato třída fakticky obsahuje pouze jednu níže zmíněnou a popsanou metodu. Poněkud nižší počet metod je důsledkem právě toho, že agenda rizik je suplována již předchozím modulem a mnohé akce řeší již dříve popsané metody modulu *RejstrikController*.

show()

Metoda, která zajišťuje získání dat nutných pro vykreslení mapy rizika. Tato data jsou předána do pohledu, který obsahuje dvě tabulky. První z nich je rejstřík rizik, který umožňuje přehledné zobrazení rizik vybraného projektu včetně všech atributů. Mapa rizik je na úrovni kódu realizována rovněž tabulkou. Pomocí CSS jsou upraveny její buňky, tak aby byla zachována myšlenka uspořádání celkové závažnosti rizik (vlevo dole nejméně závažná rizika, vpravo nahoře kritická rizika). Vzhledem ke skutečnosti, že v databázi nejsou rizika uložena ve stejné podobě, jako jsou zobrazena v rejstříku, bylo nutné rozdělit tento úkol do několika segmentů. V prvním kroku jsou vybrány rizika příslušného projektu, včetně hodnot všech atributů. Následně jsou v cyklu inkrementovány hodnoty interních proměnných, které de facto určují počet rizik, které náleží do jedné souřadnice mapy rizik. Všechny tyto proměnné jsou předány pohledu. Na Obr. 19 je vidět příklad nepříliš rozsáhlého projektu s pěti riziky a sedmi sledovanými atributy.

Rejstřík rizik projektu **Scholastic**

Název rizika	Dopad	Kategorie	Pravděpodobnost	Protipatření	Risk owner	Spouštěč	Zdroj rizika
Výpadek serverů	1	Technické	1	Zajištění kvalitního webhostingu, možné zrcadlení serverů	dusan@valec.cz	Nedostatek financí vynaložených na serverovnu, výběr nekvalitního hostingu, celkový nízký rozpočet projektu	Serverovna
Změna vedení	3	Lidské	1	-	dusan@valec.cz	Neshody ve vedení společnosti, možnost výměny klíčového zaměstnance, který nebude mít zájem na dalším trvání projektu.	management
Nedostatek prostředků	1	Procesní	2	Správně navrhnutí, odhadnout a sestavit rozpočet projektu. Existence finančního polštáře.	tom@hynek.cz	Nízký základní rozpočet, špatné stanovení rozpočtu, které se projeví již v úvodních fázích projektu	zadavatel projektu
Změna termínu	4	Zákaznické	2	Smluvní zajištění. Datum musí být předem dohodnutno a odsouhlaseno oběma stranami.	dusan@valec.cz	Zákazník již od začátku trvání projektu působí dojmem, že chce uspíšit deadline projektu a tlačí na dodavatelskou firmu	zadavatel projektu
Výměna vlastníka projektu	1	Zákaznické	5	Smluvně. Pokud zadavatel od projektu odstoupí, mělo by dojít k proplacení předem dohodnutých finančních závazků.	tom@hynek.cz	Těžko odhadnutelné, jedná se o interní záležitost zadavatelské společnosti.	Tomáš Hynek

DOPAD	0	0	0	0	0
	0	1	0	0	0
	1	0	0	0	0
	0	0	0	0	0
	1	1	0	0	1
PRAVDĚPODOBNOST					

Obr. 19: Příklad rejstříku a mapy rizik (zdroj: vlastní)

4.3.7 Modul SwotController

Práci se SWOT analýzami projektů zajišťuje řadič pojmenovaný SwotController. Paleta jeho metod reflektuje základní případy užití, které mohou nastat. Snímek základní obrazovky je na Obr. 20.

index()

Funkce, která obsahuje databázový požadavek pro získání výpisu projektů uživatele. Jde tedy opět o kolekci, která je vrácena do pohledu. Při vytváření projektu jsou automaticky založena i prázdná SWOT analýza (na úrovni databáze jsou tedy vytvořena položka v tabulce swot s prázdnými poli faktorů). Uživatel tedy vidí i projekty, které ještě nemají analýzu zpracovanou nemají.

show()

Pomocí této funkce jsou načteny položky SWOT analýzy konkrétního projektu (silné stránky, slabé stránky, příležitosti, hrozby). Tyto jsou v pohledu zobrazeny do formulářových prvků typu textarea.

update()

Možnost úpravy, resp. tvorby obsahu SWOT analýz mají pouze proj. manažeři (role projmanazer). Z HTTP požadavku jsou načtena data a v databázi jsou upraveny příslušné záznamy.

Zvolte projekt pro úpravu

Projekt

- Scholastic
- LA IT Point
- Informační systém pro SOU

SWOT analýza

projekt: Scholastic

popis projektu: V rámci tohoto projektu se zaměřujeme na tvorbu IS pro učitele ZŠ Střední

S: Zřizovatel ZŠ má finanční prostředky vyhrazeny pro tento projekt, nehrozí neplacení závazků;
kontaktní osoba zadavatele má IT vzdělání;
přesně specifikované zadání projektu;

W: relativně nízký rozpočet;

O: Dobré vztahy s vedením ZŠ;
potenciální reklama;
potenciální navázání partnerství;
v případě úspěchu projektu - potenciální nové zakázky pro další ZŠ/SŠ

T: nepochopení tužeb zákazníka;
brzký deadline;
nezkušenost se zakázkami ze státního sektoru

Upravit

Obr. 20: Obrazovka pro úpravu SWOT analýzy (zdroj: vlastní)

4.3.8 Modul UserController

Slouží pro zajištění agendy uživatelských účtů. Umožňuje vytvářet a mazat účty. Jedná se o funkcionalitu, ke které má přístup pouze uživatel s rolí admin. Níže následuje popis všech tří metod třídy.

index()

Funkce, která vrací do pohledu seznam všech uživatelů. V pohledu jsou poté zobrazeni pomocí tabulky, která obsahuje e-mail a roli uživatele.

create()

Pro vytváření nových uživatelských účtů je k dispozici formulář, data z formuláře jsou odeslána metodou PATCH a pomocí této metody zpracovány. Výsledkem je nový záznam v tabulce uživatelů. Tento nový uživatel může být okamžitě přiřazen k projektu a následně si může zobrazit další projektové informace.

destroy()

Funkce pro odstranění uživatelského účtu z databáze. Je provedeno jak odstranění záznamu z tabulky uživatelů, tak odstranění uživatele z účasti na projektu. V rámci pohledu je implementováno dialogové okno implementované pomocí Javascriptu, které by mělo zamezit nechtěnému smazání.

4.3.9 Modul UserTeamController

Jedná se o modul, který spravuje působení členů v týmech. Opět jde o funkcionalitu, kterou má k dispozici pouze administrátorský účet aplikace. Níže jsou popsány metody třídy.

index()

Typická indexová funkce. Zajišťuje naplnění pohledu správnými daty. Jednotlivé informace jsou uloženy celkem ve třech tabulkách: `projekty_a_clenove`, `users` a `projekt`. Prvním krokem je tedy vnitřní spojení tabulek (pomocí `INNER JOIN`) a výběr relevantních atributů. Následuje získání seznamu uživatelů a proj. manažerů z databáze. A nakonec jsou tato data ve formě kolekce odeslána do pohledu.

show()

Funkce pro zobrazení detailu projektu. Související pohled je na Obr. 15. Je rozdělen do tří sloupců. V levém je správa uživatelů, tedy možnost přidat a odebrat členy týmu. V pravém je k dispozici šablona rejstříku s možností modifikace a uprostřed je možnost změny proj. manažera. Nad těmito třemi sloupci je název a popis projektu s tlačítkem smazat. Všechny tyto tři prvky musí být naplněny relevantními informacemi. Je tedy nutné zajistit aby v roletkovém menu pro přidání uživatele byly na výběr pouze uživatelé, kteří k projektu ještě zapsaní nejsou. Tohoto požadavku je dosaženo aplikací metody `diff()` na kolekci členů týmu, která provede diferenci (rozdíl) dvou kolekcí. Analogicky je toto vyřešeno u výběru atributů a proj. manažera.

insert(), delete() a update_manager()

Jedná se o velmi podobné metody, které slouží pro přidání, resp. smazání uživatele z projektu. Funkce `update_manager()` zařizuje přiřazení nového proj. manažera. Ve všech případech je provedeno ověření uživatele (popsáno v kapitole 4.3.10), zda je pověřen administrátorskou rolí. Poté následuje samotné provedení databázové operace a přesměrování zpět na detail projektu, kde jsou již uživateli zobrazena aktualizovaná data.

4.3.10 Modul NeopravenyPristupController

Jde o jednoduchý řadič, který pro administrátorský účet pouze vypisuje hodnoty z tabulky `neopraveny_pristup`. Modul tedy obsahuje pouze typickou funkci `index()`.

`index()`

Po ověření zda je požadavek odeslán z administrátorského účtu, jsou spojením tabulek získány potřebné informace a ty jsou předány do pohledu, který je zobrazí ve formě tabulky.

Zajímavější je funkce, která plní tabulku daty, resp. ověřuje zda má uživatel pravomoc pro provedení určité akce (smazání, modifikaci, zobrazení). Tato je v různých obměnách využita napříč celou implementací.

```
protected function autor ($id)
{
    $je_zapsan = \App\UserTeam::where('login',auth()->user()->email)->where('idproj',$id)->first();
    if(empty($je_zapsan))
    {
        $row = new \App\neopravenyPristup;
        $row->idproj = $id;
        $row->email = auth()->user()->email;
        $row->detaily = "pokus o přístup do detailu projektu";
        $row->save();
        return true;
    }
    return false;
}

.
.
.

if(self::admin())
{
    return redirect()->route('proj') ->with('alert', 'Uživatel není administrátor.\nPokus o přístup
    byl zaznamenán.');
```

Obr. 21: Autorizační metoda řadiče (zdroj: vlastní)

Na Obr. 21 je ukázka kódu metody, která zjišťuje, zda je právě přihlášený uživatel zapsaný ke konkrétnímu projektu. Tato funkce může být volána v rámci řadiče několikrát, pokaždé když je třeba tuto skutečnost ověřit. Typicky na začátku každé metody. V tomto konkrétním případě je ověřeno, zda se záznam s odpovídajícím složeným primárním klíčem nachází v příslušné tabulce. Pokud ne, je vytvořen nový záznam v tabulce `neopraveny_pristup` a je vrácena hodnota `true`, což ústí v přesměrování uživatele a zobrazení hlášky pomocí dialogového okna. V opačném případě je vrácena hodnota `false` a volající metoda dále pokračuje v provádění příkazů.

5 Testování

Smyslem testovací fáze je objevení nedostatků a chyb aplikace, které by měly být odstraněny pro dosažení vyšší uživatelské přívětivosti a funkčnosti. Testování bylo rozděleno do dvou částí, které probíhaly sériově za sebou. V průběhu implementace jsem použil ladící program *Laravel Telescope* [17], který není součástí standardního instalačního balíku, ale lze jej zdarma stáhnout a doinstalovat.

5.1 Verifikační testování

V rámci verifikačního testování, které jsem prováděl sám při vývoji aplikace, jsem ověřoval, zda je naplněna funkcionality aplikace. Tento způsob testování je aplikován soustavně během vývoje.

Grafické uživatelské rozhraní

Při testování GUI jsem se snažil vžít do role uživatele, který aplikaci vidí poprvé a snaží se o co nejrychlejší a nejefektivnější korektní provedení zamýšleného úkonu. Již před implementací jsem si na papíře načrtnul drátové modely aplikace, při jejichž tvorbě jsem vycházel z podoby jiných aplikací podobného účelu. Při realizaci aplikace jsem se snažil dodržet to, co jsem si předem stanovil, ale když jsem si takto vytvořené rozhraní podle původních náčrtů procházel, v mnoha případech se mi stalo, že ten či onen prvek je umístěn nevhodně a neintuitivně. Tyto pokusy vedly k výsledné podobě aplikace, kterou jsem předal dále na uživatelské testování.

Back-end aplikace

Pochopitelně náročnější bylo testování funkcionality aplikace ve smyslu aplikační logiky, tedy back-endu. V praxi to probíhalo tak, že po implementaci každé jednotlivé metody jsem aplikaci spustil na serveru a procházel možné případy užití. Právě v této fázi jsem velmi často využíval rozhraní *Telescope*, které umožňuje např. zobrazit obsah jednotlivých požadavků včetně odpovědi (*response*). Toto je velmi praktický postup, který je poněkud elegantnější než vrácení obsahu pomocí `return`. Další funkci, kterou jsem využil (z ladícího rozhraní *Telescope*) bylo zobrazení těla SQL dotazů, které jsou zasílány pomocí *Eloquent* na server MySQL.

5.2 Uživatelské testování

Důvodem proč jsem v testovací fázi aplikoval uživatelské testování je, že člověk, který aplikaci a její rozhraní vidí denně (já), není schopen odhalit nedostatky, které může spatřit tester, který přijde s aplikací do styku poprvé. Je obecně známý fakt, že každý člověk vykazuje podobné vzorce chování, které se postupem času nemění. Pokud bych tuto myšlenku měl vztáhnout k užívání programu, je obvyklé, že některé akce děláme podvědomě stále dokola velice podobně, tedy posloupnost kroků při testování aplikace je víceméně stejná. Pokud navíc testování provádí sám autor kódu, jenž ví jak je který modul programu napsán, je pravděpodobnost pochybení při testování ještě vyšší.

Z těchto důvodů je výhodné provádět uživatelské testování. Ještě před započítím tohoto procesu jsem musel vyřešit několik klíčových otázek. První spočívala v tom ujasnit si, jaké bude složení testerů. Vzhledem k charakteru aplikace by se mělo jednat o lidi, kteří jsou vzdělaní, mají praxi, nebo pracují v oblasti informačních technologií a zároveň ve svém zaměstnání zastávají manažerskou funkci. Aplikace je ovšem koncipována tak, že poskytuje

základní rozhraní pro řízení *jakéhokoliv* projektu, ne nutně softwarového. Tento fakt je odrazem toho, že PMI se nezaobírá pouze sférou IT, nýbrž obecně, řízením projektů. A protože program je vytvořen na základě specifikace požadavků založených právě na zmíněné publikaci, je možné jej aplikovat např. ve strojírenství. Je pravda, že například problematika kontrolních seznamů, je rozpracována primárně pro doménu IT, nicméně je možné doplnit otázky tak, aby byly relevantní i pro jiné oblasti zájmu. Co se týče počtu testerů, existují studie, které zjistily, že již 5 testerů je schopno odchytnout 75% zásadních problémů systému [18].

Profily testerů

Pro uživatelské testování jsem se snažil vybrat ze svého okolí lidi, kteří mají zkušenosti s IT (vzdělání, práce), pracují v managementu nebo u nich alespoň existuje šance, že by s podobným systémem mohli přijít do styku. Dá se říci, že požadavky na testery vyplývají z předchozího odstavce. Konkrétní profily testerů jsou v Tab. 3.

TESTER	POHLAVÍ	VĚK	POVOLÁNÍ (POZICE)	VZDĚLÁNÍ (ÚROVEŇ)
A	muž	26	Android developer	oblast IT (VŠ)
B	žena	23	finanční administrátor	oblast financí (VŠ)
C	muž	25	C# developer	oblast IT (VŠ)
D	muž	26	senior PHP developer (vedoucí vývoje)	oblast IT (VŠ)
E	muž	24	regionální manager	oblast financí (VŠ)
F	muž	40	senior technik vysílacích služeb	oblast telekomunikací (SŠ)

Tab. 3: Profily testerů (zdroj: vlastní)

Samotné testování probíhalo tak, že jsem testerům nejdříve vysvětlil základní principy aplikace a následně jsem jim postupně zadával úkoly, které mají plnit. Při jejich provádění jsem sledoval jejich počínání a dělal si poznámky. Seznam úkolů včetně úspěšnosti úkol splnit jsem zanesl do Tab. 4. Pro popis jednání uživatele v rámci každého z testů jsem využil následující škálu:

- 1 – uživatel není schopen úkol splnit,
- 2 – uživatel má znatelné problémy při plnění úkolu, je nutné mu lehce poradit,
- 3 – uživateli splnění úkolu chvíli trvá, ale je schopen ho dokončit bez cizího zásahu,
- 4 – uživatel nemá problémy při provádění úkolu.

Důležitým poznatkem je, že žádný z testujících uživatelů nepřišel do styku s aplikací až do doby započetí testování. Tímto způsobem jsem se snažil zachytit reálnou situaci, kdy uživatel poprvé vstoupí do prostředí webové aplikace a je nucen se sám zorientovat v obsahu a plnit zamýšlené akce.

ČÍSLO TEST. PŘ.	POPIS TESTOVACÍHO PŘÍPADU	A	B	C	D	E	F	Σ
1	Vytvoření nového uživatele	4	4	4	4	4	4	24
2	Vytvoření projektu a přidání uživatele do řešitelského týmu	4	4	4	4	4	3	23
3	Vytvoření atributu šablony rejstříku rizik	3	4	4	4	4	4	23
4	Vytvoření projektu s nově vytvořeným atributem rejstříku rizik	3	4	4	4	4	4	23
5	Přidání uživatele do řešitelského týmu existujícího projektu	4	3	3	4	3	4	21
6	Smazání projektu	4	4	4	4	4	3	24
7	Smazání položky z katalogu atributů rejstříku rizik	4	4	4	4	4	4	24
8	Přidání nové otázky kontrolního seznamu do oblasti technologických rizik	4	4	4	4	4	4	24
9	Přidání (vytvoření nového rizika) u libovolného projektu	3	2	3	4	3	4	19
10	Upravení parametru Dopad u předem určeného rizika konkrétního projektu	4	4	4	4	4	4	24
11	Přidání položky do kontrolního seznamu určeného projektu	3	4	4	4	4	4	23
12	Zobrazení rejstříku rizik určeného projektu	4	4	4	4	4	4	24
13	Zobrazení SWOT analýzy určeného projektu	4	4	4	4	4	4	24

Tab. 4: Průběh uživatelského testování (zdroj: vlastní)

Emoční karty

Na závěr každého testování jsem si připravil tzv. *emoční karty*. Jedná se o prostředek pro zachycení uživatelské spokojenosti, který byl vyvinut společností Microsoft [19]. Místo toho, abych tedy pokládal otázky ve smyslu „co se vám líbilo/nelíbilo?“ jsem aplikoval tento postup. Smyslem vzniku karet je tedy možná neschopnost testera vyjádřit své pocity z použití aplikace. Tento faktor by mohl být ještě více zesílen tím, že v tomto případě se jedná o amatérské testery.

Prakticky to probíhalo tak, že jsem si podobné karty (menší množství) vyrobil a vystříhl, jednalo se pouze o černě orámované kartičky s černým textem a bílým pozadím. Měl jsem dojem, že barvy použité na kartách ve výše uvedeném odkazu by mohly vést k neobjektivním výsledkům. Následně v závěru testování jsem každého z testerů požádal ať si vybere pět karet. Mezi nejčastějšími charakteristikami se vyskytovaly následující: *nápomocné* (4krát), *intuitivní* (4krát) a *užitečné* (3krát).

5.3 Závěry testování

Emoční karty mi umožnily vžít se do pocitů uživatelů, které já, jakožto vývojář, který je v podstatě nepřetržitěm styku s aplikací, nemám šanci získat. V závěru každého testování, po emočních kartách, jsem se každého testera vyptával, který z úkolů byl pro něj nejtěžší, co by udělal jinak, které prvky uživatelského rozhraní ho nejvíce mátl, atd. Nejvíce přínosné pro mě byla zpětná vazba od testerů A, C a D.

Konkrétně tester D stojí za několika úspěšnými velkými weby nadnárodních společností. Jedná se o člověka jehož názoru si vážím. Proto jsem dal na jeho odborné rady a došlo tak k přesunu několika prvků GUI z navigační lišty menu „dovnitř“ jednotlivých obrazovek, dále jsem změnil rozložení obrazovky pro tvorbu projektů, tak aby bylo intuitivnější a estetičtější. Na základě feedbacku od testerů A a C jsem změnil zobrazení některých prvků (na úrovni CSS), které by opět mělo vést k lepší uživatelské přívětivosti.

Testovací případy 9 a 21 mají celkově nejnižší skóre úspěšnosti u uživatelů. Je to způsobeno nevhodným rozmístěním ovládacích prvků, které jsem v důsledku tohoto testování přesunul do výhodnějšího a intuitivnějšího umístění.

6 Závěr

Cílem této práce je realizace prototypu nástroje pro podporu řízení rizik. Než jsem však mohl přikročit k fázi implementace, musel jsem nastudovat velké množství informací z oblasti řízení projektu. K tomu jsem využil (dle zadání práce) standard PMI, který mi poskytl široký rozhled na téma managementu projektů. Navzdory tomu, že výsledný program by měl zajišťovat pouze agendu rizik (resp. pouze jejich identifikaci a plánování), je nutné mít povědomí i o ostatních znalostních oblastech a to z toho důvodu, že všechny aspekty řízení projektů jsou provázány. Abych na tuto problematiku nenahlížel pouze z jednoho úhlu pohledu, studoval jsem i další zdroje, které poskytují informace o aplikované teorii na praxi (např. publikace Řízení projektů v IT). Všechny tyto nově nabyté vědomosti jsem využil při specifikaci požadavků na aplikaci. Rozhodl jsem se, že v rámci získávání dat procesu identifikace rizik vytvořím v programu softwarovou podporu pro práci s kontrolními seznamy a v rámci analýzy dat podporu pro tvorbu SWOT analýzy. Tyto dva prvky by měly vytvářet pilíře pro tvorbu rejstříku rizik, který je ústředním výstupem aplikace. Ovšem je vhodné poznamenat, že jak uvádí PMI, tyto dvě techniky jsou pouze jednou z několika možných (více se tomuto tématu věnuji v kapitole 3.1) a pro naplnění rejstříku relevantními daty je vhodné je mezi sebou kombinovat. Kromě výše zmíněného aplikace poskytuje prostředky pro správu uživatelských účtů a správu projektů.

Realizace webové aplikace probíhala za použití aplikačního rámce Laravel, který využívá architekturu MVC. V tomto ohledu to pro mě byla premiéra, protože s žádným frameworkem jsem až doposud neměl zkušenosti. Implementace aplikace probíhala postupně dle modelu případů užití. V systému figurují tři aktéři, kteří se liší paletou pravomocí a akcí, které mohou v systému provádět. Administrátorský účet může vytvářet uživatelské účty, které pak lze přiřadit do řešitelských týmů projektů. Dále je zodpovědný za správu kontrolních seznamů, jejichž obsah vytváří. A v neposlední řadě vytváří projekty, kterým kromě týmu přiděluje i sledované atributy rejstříku rizik. Administrátor tedy de facto vytvoří pro projektového manažera rozhraní, na kterém může stavět a utvářet ho dle svého vlastního uvážení. Projektový manažer tedy od administrátora převezme prázdnou obálku, se kterou pracuje a plní ji daty, se kterými je možno dále pracovat. Poslední rolí je člen týmu, který může nahlížet do rejstříku rizik, prohlížet SWOT analýzu a pracovat s kontrolním seznamem.

Perzistentní data jsou uchovávána systémem řízení báze dat MySQL. V databázi jsou obsaženy otázky kontrolního seznamu, které byly převzaty od R.S. Pressman & Associates, Inc. [20] Je to z toho důvodu, aby aplikace byla blíže připravenosti na reálné nasazení.

Testování aplikace probíhalo verifikačně a posléze uživatelsky. V rámci uživatelského testování jsem obdržel cennou zpětnou vazbu od testerů, kteří se věnují na profesionální úrovni problematice vývoje webových aplikací. Tato zpětná vazba, včetně všech odhalených nedostatků byla do finální verze aplikace zapracována a vedla k poměrně značným změnám v rámci GUI.

Vzhledem k faktu, že se jedná pouze o prototyp, který poskytuje nutné funkcionality pro správu a řízení rizik projektu, je zde prostor pro *rozšíření*.

Já osobně jej spatřuji například v přidání funkce *kalendáře*, která by mohla v podstatě jakémukoliv prvku aplikace přidat další rozměr v podobě plánování úkolů. V praxi by to mohlo vypadat třeba tak, že pro zaopatření rizika je nutné vykonat sekvenci akcí, které musí být hotové do určitého data. Těchto akcí může být více, tak mohou být rozděleny mezi více osob (typicky mezi více členů týmu). Jde v podstatě o rozšíření problematiky vlastnictví rizika, tedy atributu risk-owner.

Další oblastí, kterou lze rozšířit jsou různé *aplikační podpory* pro analýzu a shromažďování daty (některé z nich byly diskutovány v kapitole 3.1). Do této aplikace byla vybrána z každé části pouze jedna položka (SWOT a kontrolní seznamy). Bylo by tedy možné přidat softwarovou podporu i pro jiné techniky a nástroje, což by mělo vést k benefitu vyšší komplexnosti systému a ve výsledku i k přesnější identifikaci rizik.

Literatura

- [1] *A guide to the project management body of knowledge / Project Management Institute*. Sixth edition. Newtown Square, PA: Project Management Institute, 2017, ISBN 978-1-62825-184-5.
- [2] Tuckman forming storming norming performing model. *Business balls* [online]. [cit. 2019-01-12]. Dostupné z: <https://www.businessballs.com/team-management/tuckman-forming-storming-norming-performing-model/>
- [3] *Řízení rizik ve firmách a jiných organizacích*. 3., rozš. a aktualiz. vyd. Praha: Grada, 2010, s. 90. Expert (Grada). ISBN 978-80-247-3051-6.
- [4] RAIS, Karel a Radek DOSKOČIL. *Risk management: studijní text pro kombinovanou formu studia*. Sixth edition. Brno: Akademické nakladatelství CERM, VUT v Brně, 2007, ISBN 978-80-214-3510-0.
- [5] *Řízení projektů v IT: kompletní průvodce*. Brno: Computer Press, 2011, ISBN 978-80-251-2882-4.
- [6] Updated PMP Exam has Launched. *PMI - Project Management Institute* [online]. [cit. 2019-01-12]. Dostupné z: <https://www.pmi.org/certifications/types/project-management-pmp/exam-prep/changes>
- [7] RAIS, Karel a Radek DOSKOČIL. *Risk management: studijní text pro kombinovanou formu studia*. Sixth edition. Brno: Akademické nakladatelství CERM, VUT v Brně, 2007, ISBN 978-80-214-3510-0.
- [8] SHRIVASTAV, Narendra K. *Project Management Institute: Project risk management--another success-boosting tool in a PM's toolkit* [online]. Paper presented at PMI® Global Congress 2012 [cit. 2019-01-14]. Dostupné z: <https://www.pmi.org/learning/library/project-risk-management-success-tool-6078>
- [9] *Laravel: The PHP Framework For Web Artisans* [online]. [cit. 2019-05-01]. Dostupné z: <https://laravel.com/>
- [10] *Laracasts: Screencasts for the Modern Developer* [online]. [cit. 2019-05-01]. Dostupné z: <https://laracasts.com/>
- [11] REIGNS, Stephanie. *11 Best PHP Frameworks for Modern Web Developers in 2019: Screencasts for the Modern Developer* [online]. [cit. 2019-05-01]. Dostupné z: <https://coderseye.com/best-php-frameworks-for-web-developers/>
- [12] *CodeIgniter: CodeIgniter Web Framework* [online]. [cit. 2019-05-01]. Dostupné z: <https://codeigniter.com/>

- [13] *Requirements analysis and system design*. 2nd ed. New York: Pearson/Addison Wesley, 2005, s. 171. ISBN 0321204646.
- [14] *Eloquent: Getting Started: Retrieving Single Models / Aggregates* [online]. [cit. 2019-05-01]. Dostupné z: <https://laravel.com/docs/5.8/eloquent#retrieving-single-models>
- [15] *Laravel: Database: Migrations* [online]. [cit. 2019-05-01]. Dostupné z: <https://laravel.com/docs/5.8/migrations>
- [16] *Bulma: Bulma is a free, open source CSS framework* [online]. [cit. 2019-05-01]. Dostupné z: <https://bulma.io/>
- [17] *Laravel Telescope* [online]. [cit. 2019-05-01]. Dostupné z: <https://laravel.com/docs/5.8/telescope>
- [18] NIELSEN, Jakob. *Why You Only Need to Test with 5 Users: User Testing* [online]. [cit. 2019-05-01]. Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [19] *Emoční karty - Card Sorting: AITOM* [online]. [cit. 2019-05-01]. Dostupné z: <https://www.aitom.cz/get.php?id=1242>
- [20] *R.S. Pressman & Associates, Inc.: Checklists: Risk Assessment* [online]. [cit. 2019-05-01]. Dostupné z: <http://www.rspa.com/checklists/risk.html>

7 Seznam příloh na CD

- `README` – návod k použití,
- `/rmt` – projekt v Laravelu,
- `/thesis` – tato technická zpráva.